Exact and approximate algorithms for computing the hyperbolicity of large-scale graphs

Nathann Cohen, David Coudert, Aurélien Lancin

RESEARCH REPORT
N° 8074
September 2012
Project-Team MASCOTTE

ISSN 0249-6399 ISRN INRIA/RR--8074--FR+ENG

Exact and approximate algorithms for computing the hyperbolicity of large-scale graphs

Nathann Cohen[*], David Coudert[†], Aurélien Lancin[‡]

Project-Team MASCOTTE

Research Report n° 8074 — September 2012 — 28 pages

Abstract: Let G be a connected graph, and let $\mathbb{d}(a,b)$ denotes the shortest path distance between vertices a and b of G. The graph G is δ-hyperbolic if for any vertices a,b,c,d of G, the two largest of the three sums $S_1 = \mathbb{d}(a,b) + \mathbb{d}(c,d)$, $S_2 = \mathbb{d}(a,c) + \mathbb{d}(b,d)$, and $S_3 = \mathbb{d}(a,d) + \mathbb{d}(b,c)$ differ by at most 2δ. This can be determined in time $O(n^4)$ which could be prohibitive for large graphs. In this document, we propose an exact algorithm for determining the hyperbolicity of a graph that is scalable for large graphs. The time complexity of this algorithm is a function of the size of the largest bi-connected component of the graph, of the shortest path distance distribution in this component and of the value of the hyperbolicity. In the worst case, the time complexity remains in $O(n^4)$, but it is much faster in practice. Indeed, it allowed us to compute the exact hyperbolicity of all maps of the autonomous systems of the Internet provided by CAIDA and DIMES. We also propose both a multiplicative factor and an additive constant approximation algorithms. Finally, we also analyze further the time complexity of our exact algorithm for several class of graphs.

Key-words: Hyperbolicity, algorithm, graph, approximation.

[*] LRI, Université Paris-Sud 11, France. `nathann.cohen@lri.fr`
[†] MASCOTTE Project, INRIA/I3S(CNRS/UNSA), France. `david.coudert@inria.fr`
[‡] MASCOTTE Project, INRIA/I3S(CNRS/UNSA), France. `aurelien.lancin@inria.fr`

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Algorithmes exact et approché pour le calcul de l'hyperbolicité d'un graphe

Résumé : Soit G un graphe connexe et soit $\mathbb{d}(a,b)$ la distance entre les sommets a et b dans le graphe. Le graphe G est dit δ-hyperbolic si pour tout quadruplet a, b, c, d de sommets dans G, les deux plus grandes des sommes $S_1 = \mathbb{d}(a,b) + \mathbb{d}(c,d)$, $S_2 = \mathbb{d}(a,c) + \mathbb{d}(b,d)$, et $S_3 = \mathbb{d}(a,d) + \mathbb{d}(b,c)$ diffèrent d'au plus 2δ. Cette valeur peut être déterminé en temps $O\left(n^4\right)$, ce qui est souvent inaccessible pour les grands graphes.

Nous proposons un nouvel algorithme exact pour calculer l'hyperbolicité sur des graphes de grande taille. La complexité en temps de cet algorithme est fonction de la taille de la plus grande composante bi-connexe du graphe, de la distribution des plus courts chemins dans cette composante et de la valeur de l'hyperbolicité. Dans le pire cas, cet algorithme prendra un temps en $O\left(n^4\right)$. Cependant, l'algorithme est bien plus efficace en pratique. Il nous a en effet permis de calculer la valeur exacte de l'hyperbolicité pour l'ensemble des cartes Internet CAIDA et DIMES. Nous proposons également un algorithme approché avec un facteur multiplicatif ou avec une constante additive donné en entrée. Enfin, nous analysons la complexité temporelle de notre algorithme pour des classes particulières de graphes.

Mots-clés : Hyperbolicité, algorithme, graphe, approximation.

1 Introduction

In the last years, extensive work have been carried out to better understand the structural properties of the Internet topology. One of the main objective of these studies is to design new routing models that could face the current growth rate of the Internet topology and improve upon the actual inter-domain routing protocol, BGP (Border Gateway Protocol), in terms of scalability. In particular, it has been shown that the Internet topology can be accurately embedded into an hyperbolic space [6] and that a simple greedy-forwarding algorithm can be expected to perform very well [25].

The (Gromov) *hyperbolicity* of a graph reflects how the metric space (distances) of a graph is close to the metric space of a tree. Gromov [18] defines the notion of δ-hyperbolic metric spaces using the notion of δ*-thin* triangles. More precisely, given any three points x, y, and z of a hyperbolic metric space, the triangle (x, y, z) is δ-thin if any point of the geodesic joining x and y is at distance at most δ of one of the geodesics joining x to z or y to z. A δ-hyperbolic space is a geodesic metric space in which every geodesic triangle is δ-thin. In other words, a graph has hyperbolicity at most δ if, for any $u, v, w \in V(G)$ and for any shortest paths P_{uv}, P_{vw}, P_{uw} between these three vertices, any vertex in P_{uv} is at distance at most δ from $P_{vw} \cup P_{uw}$. Intuitively, in a graph with small hyperbolicity, the shortest-paths joining a pair of vertices are close to each other. For instance, trees and cliques are 0-hyperbolic which reflects the uniqueness of shortest paths, while $n \times n$ grids are $n - 1$ hyperbolic since these graphs have many different shortest paths between pairs of vertices.

An alternative definition given by Gromov [18], and called the 4-*points condition*, is the following. Let $\mathbb{d}(u, v)$ denote the distance between vertices u and v. A metric space is δ-hyperbolic if for any four points u, v, w, x the two largest of the distance sums $\mathbb{d}(u, v) + \mathbb{d}(w, x)$, $\mathbb{d}(u, w) + \mathbb{d}(v, x)$, $\mathbb{d}(u, x) + \mathbb{d}(v, w)$ differ by at most 2δ. This notion extends to connected graphs and we say that a connected graph $G = (V, E)$ equipped with its standard graph metric $\mathbb{d}_G$ (shortest path distance) is δ-hyperbolic if the metric space $(V, \mathbb{d}_G)$ is δ-hyperbolic. In other words, a connected graph G is δ-hyperbolic if it satisfies the 4-points condition. The hyperbolicity of a graph measures its tree-likeness. The less the value of δ is, the more the graph looks like a tree. Moreover, it can be observed that the hyperbolicity of a connected graph is the maximum value of the hyperbolicity of its biconnected components.

From this second definition it is obvious that determining the hyperbolicity δ of a graph of order n can be done in time $O(n^4)$, by testing all 4-tuples of vertices of the graph. An implementation of the naive algorithm for determining the hyperbolicity of a graph as been included into the distory (Distance Between Phylogenetic Histories) package [13] of the CRAN (The Comprehensive R Archive Network) project [11]. This package is devoted to the study of geodesic distance between phylogenetic trees and associated functions. The implementation uses the "revolving doors Gray code" principle [22] for visiting all 4-tuples of the input graph. The time complexity has recently been improved to $O(n^{3.69})$ [17] using the fast (max,min)-matrix multiplication algorithm proposed in [15]. However, the computation time for large-scale graphs remains prohibitive. For instance, for a graph with 25 815 vertices (size of the largest bi-connected component of the last CAIDA map of the Autonomous Systems of the Internet [16]), the algorithm has to iterate over around $1.85\ 10^{16}$ 4-tuples which represents several weeks of computations.

Also, a heuristic algorithm for determining the hyperbolicity of CAIDA AS maps has been used in [12], and a 2-approximation algorithm, with running time in $O(n^3)$, is obtained fixing one vertex and evaluating all possible 4-tuples containing that vertex [10]. Clearly, faster exact and approximate algorithms are needed. Furthermore, these algorithms would be useful in other research fields where the hyperbolicity property is used such as network security [20, 21], traffic flow [24] and phylogenetics [14].

The need for more efficient algorithms has been expressed by many authors, for instance it has recently been mentioned in the conclusion of [9] that "*exact computation of δ by its definition takes $O(n^4)$ time, which is not scalable to large graphs, and thus the design of more efficient exact or approximation algorithms would be of interest*".

Our results

In this paper, we present a new exact algorithm for computing the hyperbolicity of a graph. It is the first exact algorithm scalable for large graphs. Indeed, applied to the latest CAIDA map (2012/06/01), we have been able to reduce the computation time from 24 expected days with the naive algorithm to 8 days with our algorithm. Furthermore, the proposed algorithm can be turned into an approximation algorithm with tunable multiplicative factor or additive constant. While the worst case time complexity of the proposed algorithm is $O(n^4)$, it is much faster in practice.

We start in Section 2 with some definitions and notations. Then we formally describe our algorithm for computing the hyperbolicity of a graph in Section 3 and give hints on its time complexity. We also explain how to turn this algorithm into an approximation algorithm. Then we show in Section 4 how to use a partition by clique-separators to speed-up the running time of the algorithm. In Section 5, we evaluate experimentally the performances of our exact algorithm. Next, in Section 6, we report on the computation of the hyperbolicity of large-scale graphs, namely CAIDA and DIMES maps of the Internet autonomous systems topology. Last, we list some simple results and proofs on the hyperbolicity of particular graph classes in Appendix A.

2 Definitions and known results

In this section, we first recall the definition of the hyperbolicity of a graph and fix some notations. Then, we survey some known results, in particular the value of the hyperbolicity of some families of graphs, and general lower and upper bounds.

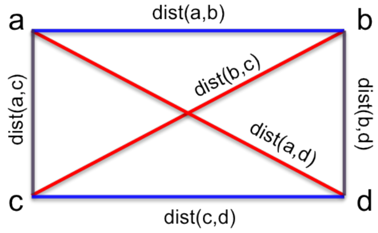
Let G be a connected graph, and let $\mathbb{d}(a,b)$ denotes the shortest path distance between vertices a and b of G. The hyperbolicity of a graph has been defined by Gromov [18] as follows.

Definition 1 ([18])**.** *The graph G is δ-hyperbolic if for any vertices a,b,c,d of G, the two largest of the three sums $S_1 = \mathbb{d}(a,b) + \mathbb{d}(c,d)$, $S_2 = \mathbb{d}(a,c) + \mathbb{d}(b,d)$, and $S_3 = \mathbb{d}(a,d) + \mathbb{d}(b,c)$ differ by at most 2δ.*

Figure 1 illustrates Definition 1 also called 4-*points condition*. In the following, we denote $\mathrm{h}(a,b,c,d)$ the difference between the two largest of the three sums S_1, S_2, and S_3, and so we have $2\delta = \max_{a,b,c,d\in V} \mathrm{h}(a,b,c,d)$.

In this paper, we consider unweighted graphs only. Therefore, the possible values of δ are half-integers, that is $\delta \in \left\{\frac{i}{2},\ i \in \mathbb{N}\right\}$. Furthermore $\delta \leq \frac{D}{2}$, where D is the diameter of the graph. This result also follows from Lemma 5 that we will prove in next section.

The hyperbolicity of a graph is the maximum of the hyperbolicity of its biconnected components. To show this, let x be a cut-vertex of the graph G and w.l.o.g. let B_1 and B_2 be the two components of G separated by x. Let now $a,b,c \in B_1$ and $d \in B_2$. We have $S_1 = \mathbb{d}(a,b)+\mathbb{d}(c,d) = \mathbb{d}(a,b)+\mathbb{d}(c,x)+\mathbb{d}(x,d)$, $S_2 = \mathbb{d}(a,c)+\mathbb{d}(b,d) = \mathbb{d}(a,c)+\mathbb{d}(b,x)+\mathbb{d}(x,d)$, and $S_3 = \mathbb{d}(a,d)+\mathbb{d}(b,c) = \mathbb{d}(a,x)+\mathbb{d}(x,d)+\mathbb{d}(b,c)$. The computed value for the 4-tuple a,b,c,d is the same than for the 4-tuple a,b,c,x (See Fig. 2a). Similarly, when $a,b \in B_1$ and $c,d \in B_2$, the computed value will always be 0 (See Fig. 2b).



$S_1 = \mathrm{d}(a,b) + \mathrm{d}(c,d)$
$S_2 = \mathrm{d}(a,c) + \mathrm{d}(b,d)$
$S_3 = \mathrm{d}(a,d) + \mathrm{d}(b,c)$

If $S_1 \geq S_2 \geq S_3$, then $\delta \leq \frac{S_1 - S_2}{2}$

Figure 1: The 4-*points condition*.

(a) $\mathrm{h}(a,b,c,d) = \mathrm{h}(a,b,c,x)$

(b) $\delta = 0$

Figure 2: Hyperbolicity over biconnected components

For some families of graphs it is possible to give either the exact value of their hyperbolicity. In particular, we have:

- *Block* graphs (i.e., connected graph in which every 2-connected subgraph is a clique) are 0-hyperbolic, and so are trees and cliques;
- Cycles of order $n = 4p + \varepsilon$, with $p \geq 1$ and $\varepsilon \in \{0, 1, 2, 3\}$, are $(p - 1/2)$-hyperbolic when $\varepsilon = 1$, and p-hyperbolic otherwise;
- $n \times m$ grids, with $2 \leq n \leq m$, are $(n-1)$-hyperbolic;
- d-dimensional grids of side s are $(s-1)\lfloor d/2 \rfloor$-hyperbolic (see proof in Appendix A);
- The hypercube $\mathcal{H}_2(n)$ of dimension n and order 2^n is $\lfloor n/2 \rfloor$-hyperbolic (see proof in Appendix A);
- Chordal graphs are (≤ 1)-hyperbolic and 1-hyperbolic chordal graphs have been characterized [5, 7, 23];
- k-chordal graphs with $k \geq 4$ are $(\leq \lfloor k/4 \rfloor)$-hyperbolic [29].

Some theoretical bounds according to other graph parameters have also been established. Let $g(G)$ denotes the *girth* of a graph (i.e., length of the shortest cycle of the graph). We have $\left\lfloor \frac{g(G)}{4} \right\rfloor - \varepsilon_g \leq \delta$, where $\varepsilon_g = \frac{1}{2}$ when $g(G) \equiv 1 \pmod 4$ and 0 otherwise. An upper bound has been established similarly in [8] using the *circumference* $c(G)$ of the graph, that is the maximum length of its cycles. A tighter upper bound follows from the chordality $k(G)$ of the graph since for any k-chordal graph we have $k(G) \leq c(G)$. Summarizing,

Theorem 2. *For every graph G, we have $\left\lfloor \frac{g(G)}{4} \right\rfloor - \varepsilon_g \leq \delta(G) \leq \left\lfloor \frac{k(G)}{4} \right\rfloor$, where $\varepsilon_g = \frac{1}{2}$ when $g(G) \equiv 1 \pmod 4$ and 0 otherwise.*

Other bounds on the hyperbolicity of a graph have been established. Let $\beta(G)$ be the *independence number* of G (i.e., cardinality of the largest independent set) and let $\gamma(G)$ be the *domination number* of G (i.e., size of the smallest dominating set). We have,

Theorem 3 ([26]). *For every graph G of order n, we have $\delta(G) \leq \min\left\{\beta(G),\ \frac{n-\beta(G)+2}{2}\right\}$.*

Theorem 4 ([26]). *For every graph G, we have $\delta(G) \leq \frac{3}{2}\gamma(G)$.*

3 Exact algorithm for computing the hyperbolicity

In this section, we formally describe a new exact algorithm for computing the hyperbolicity of graphs. We then give some hints on its time-complexity and explain how to turn this algorithm into an approximation algorithm.

3.1 The algorithm

Our algorithm is based on the following lemma,

Lemma 5. *Let $G = (V, E)$ be a connected graph, let $a, b, c, d \in V$, let $S_1 = \mathbb{d}(a, b) + \mathbb{d}(c, d)$, $S_2 = \mathbb{d}(a, c) + \mathbb{d}(b, d)$, and $S_3 = \mathbb{d}(a, d) + \mathbb{d}(b, c)$, and assume w.l.o.g. that $S_1 \geq \max\{S_2,\ S_3\}$. We have*

$$\mathrm{h}(a, b, c, d) \leq \min\{\mathbb{d}(a, b),\ \mathbb{d}(c, d)\}$$

Proof. We have $S_2+S_3 = \mathbb{d}(a,c)+\mathbb{d}(b,d)+\mathbb{d}(a,d)+\mathbb{d}(b,c) = (\mathbb{d}(a,c) + \mathbb{d}(b,c))+(\mathbb{d}(a,d) + \mathbb{d}(b,d))$. Using the triangular inequality, we deduce $S_2 + S_3 \geq 2 \cdot \mathbb{d}(a, b)$. Since S_1 is the largest sum, we have $\mathrm{h}(a, b, c, d) = S_1 - \max\{S_2,\ S_3\} \leq S_1 - (S_2 + S_3)/2 \leq S_1 - \mathbb{d}(a, b) = \mathbb{d}(c, d)$. We obtain similarly that $\mathrm{h}(a, b, c, d) \leq \mathbb{d}(a, b)$. □

In order to make good use of Lemma 5, we adapt the naive $O(n^4)$ algorithm so that it first tests the 4-tuples which are the most likely to yield a large hyperbolicity. More precisely, we iterate over the 4-tuples in such a way that a, b, c, d is tested before a', b', c', d' if $\min(\mathbb{d}(c, d), \mathbb{d}(a, b)) > \min(\mathbb{d}(c', d'), \mathbb{d}(a', b'))$. Assuming that `pairs` is the list of the $\binom{n}{2}$ pairs of vertices sorted decreasingly (we naturally ignore 4-tuples such that $d(a, b) \leq d(c, d)$), this yield Algorithm 1.

Thanks to Lemma 5 we know that at any step of the algorithm the value of $\mathbb{d}(c, d)$ is an upper bound on the value of $\mathrm{h}(a, b, c, d)$. If the current lower bound is h^*, none of the 4-tuples such that $\mathbb{d}(c, d) \leq \mathrm{h}^*$ can be used to improve the lower bound. We can thus cut exploration. For instance, if the input graph is a $n \times n$ grid, with diameter $2n-2$ and hyperbolicity $\delta = n-1$ (so $\mathrm{h}^* = 2n-2$), the value of the hyperbolicity will be obtained with the first considered 4-tuple. On the other hand, if the input graph is a $n \times 2$ grid, with diameter n and hyperbolicity $\delta = 1$ (so $\mathrm{h}^* = 2$), almost all 4-tuples will be considered.

Since we have $\binom{n}{2}$ pairs of vertices, and since Algorithm 1 considers pairs of pairs, the worst case time complexity of the algorithm is in $O(n^4)$ (and has a quadratic memory usage). However, we observe that we can parameterize the time complexity with the optimal value of the hyperbolicity and the distribution of the path lengths. To do so, we use the set $P[\ell]$ of pairs (a, b) such that $\mathbb{d}(a, b) = \ell$ to reformulate the algorithm with Algorithm 2.

Algorithm 1 Hyperbolicity

Require: $G = (V, E)$ is a 2-connected graph.
Ensure: δ, the hyperbolicity of G (observe that $2\delta = \mathrm{h}^*$).

1: Let `pairs` be the list of the $\binom{n}{2}$ pairs of vertices sorted by decreasing distances
2: Let $\mathrm{h}^* := 0$
3: **for** $0 \le i < \binom{n}{2}$ **do**
4:   $(c, d) :=$ `pairs`$[i]$
5:   **if** $\mathbb{d}(c, d) \le \mathrm{h}^*$ **then**
6:     break
7:   **for** $0 \le j < i$ **do**
8:     $(a, b) :=$ `pairs`$[j]$
9:     $\mathrm{h}^* := \max\{\mathrm{h}^*, \mathrm{h}(a, b, c, d)\}$
10:     **if** $\mathbb{d}(a, b) \le \mathrm{h}^*$ **then**
11:       break
12: **return** $\mathrm{h}^*/2$

Algorithm 2 Hyperbolicity (parameterized version)

Require: $G = (V, E)$ is a 2-connected graph.
Ensure: δ, the hyperbolicity of G (observe that $2\delta = \mathrm{h}^*$).

1: Let $P[\ell]$ be the list of pairs $(a, b) \in V \times V$ such that $\mathbb{d}(a, b) == \ell$
2: Sort $P[\ell]$ in lexicographic order
3: Let $\mathrm{h}^* := 0$
4: **for** $\ell_1 := D$ down to 1 **do**
5:   **for** $\ell_2 := D$ down to ℓ_1 **do**
6:     **for all** $(a, b) \in P[\ell_1]$ **do**
7:       **for all** $(c, d) \in P[\ell_2]$ **do** {When $\ell_1 == \ell_2$, we ensure that $(a, b) < (c, d)$}
8:         $\mathrm{h}^* := \max\{\mathrm{h}^*, \mathrm{h}(a, b, c, d)\}$
9:         **if** $\mathrm{h}^* == \ell_1$ **then**
10:           **return** $\mathrm{h}^*/2$

Proposition 6. *Given a δ-hyperbolic graph of diameter D and the sets $P[\ell]$ of pairs of vertices at distance ℓ from each other, the time complexity of lines 4-10 of Algorithm 2 is in*

$$O\left(\sum_{\ell_1=2\delta}^{D} |P[\ell_1]| \left(\frac{|P[\ell_1]| - 1}{2} + \sum_{\ell_2=\ell_1+1}^{D} |P[\ell_2]|\right)\right)$$

Proof. Since Algorithm 2 uses Lemma 5, it considers only pairs of pairs $((a, b), (c, d))$ such that $D \ge \mathbb{d}(c, d) = \ell_2 \ge \mathbb{d}(a, b) = \ell_1 \ge 2\delta$. Furthermore, it uses the ordering of $P[\ell]$ to test only the $\binom{|P[\ell]|}{2}$ pairs of pairs such that $\mathbb{d}(a, b) = \mathbb{d}(c, d) = \ell$. The result follows. □

Typically, for the $n \times n$ grid of diameter $2n-2$ and hyperbolicity $n-1$, we have $|P[2n-2]| = 2$ and so a single pair of pairs such that $\mathbb{d}(a, b) = \mathbb{d}(c, d) = 2n-2$. The result is obtained with the first considered 4-tuple, and then the exploration is stopped and the result is returned. Chordal graphs, which have hyperbolicity at most 1 [7], are worst case instances for this algorithm since its running time increases with the gap between the diameter and the hyperbolicity.

Algorithm 2 also requires to compute the distances between all pairs of vertices (line 1) and to sort these pairs by decreasing length (line 2). These operations are respectively in time $O(m(n+m))$ and $O(n^2 \log n)$ for a graph with n vertices and m edges.

Computational complexity of Algorithm 2 depends on the distance distribution of the graph and of the computed value of the hyperbolicity. We provide a close formula on its time complexity for some particular graph classes in Appendix A. Furthermore, we show in Section 5 some experimental results which highlight the running time improvements on large-scale graphs.

3.2 Turning Algorithm 2 into an approximation algorithm

Observe that at any step of the algorithm, $\mathbb{d}(a,b)/2 = \ell_1/2$ is an upper bound for the hyperbolicity of G, and $\mathrm{h}^*/2$ a lower bound. Therefore, by stopping its execution after a given time and returning the values h^* and ℓ_1, Algorithm 2 is turned into an approximation algorithm with approximation ratio $\frac{\ell_1}{\mathrm{h}^*}$. More precisely, we can insert one of the following test after line 10 of the algorithm:

- "If computation time is larger than allowed computation time, then stop computations and return h^* and ℓ_1". We get $\frac{\mathrm{h}^*}{2} \leq \delta \leq \frac{\ell_1}{2}$;
- "If $\frac{\ell_1}{\mathrm{h}^*} \leq apx$, then stop computations and return $\frac{\mathrm{h}^*}{2}$." We get an approximation of the value δ of the hyperbolicity with proven approximation factor *apx* (i.e., $\frac{\mathrm{h}^*}{2} \leq \delta \leq apx \cdot \frac{\mathrm{h}^*}{2}$);
- "If $\frac{\ell_1}{2} - \frac{\mathrm{h}^*}{2} \leq apx$, then stop computations and return $\frac{\mathrm{h}^*}{2}$." We get an approximation of the value δ of the hyperbolicity with proven additive approximation constant *apx* (i.e., $\frac{\mathrm{h}^*}{2} \leq \delta \leq \frac{\mathrm{h}^*}{2} + apx$).

As we show in the next section, the main part of the running time of the algorithm consists in closing the small gap between lower bounds, that are generally found very quickly, and upper bounds. Depending on the expected result, it may be appropriate to use the above rules that may allow to save time while preserving a sufficient precision.

4 Decomposition

In Section 2 we have seen that the hyperbolicity of a graph is the maximum of the hyperbolicity of its biconnected components. In this section, we will show that we can further reduce the problem size using a decomposition of the graph by *clique-separators* [28]. We then show the benefit of this method on Internet-like graphs.

Recall that a set $X \subset V$ is a separator of a connected graph $G = (V, E)$ if the induced subgraph $G[V \setminus X]$ has at least two distincts connected components. If furthermore the induced subgraph $G[X]$ is a clique, then X is a clique-separator for G. Let us now show that the hyperbolicity of any 4-tuples with vertices on different parts of $G[V \setminus X]$ is at most 1 when $G[X]$ is a clique.

Lemma 7. *Let $G = (V, E)$ be a connected graph, let $X \subset V$ be a clique-separator for G, and let $A, B \subset V$ be two parts of G separated by X. Then for any $a_1, a_2 \in A$ and $b_1, b_2 \in B$, we have $\delta(a_1, a_2, b_1, b_2) \leq 1$.*

Proof. Let $S_1 = \mathbb{d}(a_1, a_2) + \mathbb{d}(b_1, b_2)$, $S_2 = \mathbb{d}(a_1, b_1) + (a_2, b_2)$, and $S_3 = \mathbb{d}(a_1, b_2) + \mathbb{d}(a_2, b_1)$.

We first consider the case where $S_1 \geq S_2 \geq S_3$. Let $s_1 \in X$ be such that $\mathbb{d}(a_1, b_1) = \mathbb{d}(a_1, s_1) + \mathbb{d}(s_1, b_1)$ and let $s_2 \in X$ be such that $\mathbb{d}(a_2, b_2) = \mathbb{d}(a_2, s_2) + \mathbb{d}(s_2, b_2)$. In other words, s_1 (resp. s_2) is a vertex in X that lies on a shortest path between a_1 and b_1 (resp. a_2 and b_2). Clearly, we have the following relations:

$$\begin{aligned} \mathbb{d}(a_1, a_2) &\leq \mathbb{d}(a_1, s_1) + 1 + \mathbb{d}(s_2, a_2) \\ \mathbb{d}(b_1, b_2) &\leq \mathbb{d}(b_1, s_1) + 1 + \mathbb{d}(s_2, b_2) \end{aligned}$$

From these relations, we get:

$$\begin{aligned} S_2 &= \mathbb{d}(a_1, b_1) + (a_2, b_2) \\ &= \mathbb{d}(a_1, s_1) + \mathbb{d}(s_1, b_1) + \mathbb{d}(a_2, s_2) + \mathbb{d}(s_2, b_2) \\ &\geq \mathbb{d}(a_1, a_2) + \mathbb{d}(b_1, b_2) - 2 = S_1 - 2 \end{aligned}$$

We conclude that $\delta(a_1, a_2, b_1, b_2) \leq (S_1 - S_2)/2 \leq 1$. Similarly, we obtain that $\delta(a_1, a_2, b_1, b_2) \leq 1$ when $S_1 \geq S_3 \geq S_2$.

Now, when $S_2 \geq S_3 \geq S_1$ we use:

$$\begin{aligned} \mathbb{d}(a_1, s_2) &\leq \mathbb{d}(a_1, s_1) + 1 \\ \mathbb{d}(a_2, s_1) &\leq \mathbb{d}(a_2, s_2) + 1 \end{aligned}$$

and we get

$$\begin{aligned} S_2 &= \mathbb{d}(a_1, b_1) + (a_2, b_2) \\ &= \mathbb{d}(a_1, s_1) + \mathbb{d}(s_1, b_1) + \mathbb{d}(a_2, s_2) + \mathbb{d}(s_2, b_2) \\ &\leq \mathbb{d}(a_1, b_2) + \mathbb{d}(a_2, b_1) + 2 = S_3 + 2 \end{aligned}$$

We conclude that $\delta(a_1, a_2, b_1, b_2) \leq (S_2 - S_3)/2 \leq 1$. □

Using similar arguments than in the proof of Lemma 7, we can show that 4-tuples with one vertex in one part A and the other vertices in a different part B separated by the clique $G[X]$ from A have hyperbolicity at most 1.

Lemma 8. *Let $G = (V, E)$ be a connected graph, let $X \subset V$ be a clique-separator for G, and let $A, B \subset V$ be two parts of G separated by X. Then for any $a \in A$ and $b_1, b_2, b_3 \in B$, we have $\delta(a, b_1, b_2, b_3) \leq 1$.*

From above results, we deduce that

Corollary 9. *Let $G = (V, E)$ be a connected graph, let X be a clique-separator for G, and let $A_i \subset V$ with $i \geq 2$ be the parts of G separated by X. We have*

$$\delta(G) \leq \max\left\{1, \max_{i=1,2,\cdots k} \delta(G[A_i \cup X])\right\}$$

Recall that we can decide in linear time if the graph has hyperbolicity at most 1 [5, 7, 23]. Therefore, when we know that the hyperbolicity is at least 1, we use Corollary 9 to decompose the graph into smaller parts on which to compute the hyperbolicity. Moreover, above results can be used recursively if some graph $G[A_i \cup X]$ also has a clique-separator. The complete decomposition of the graph by clique-separators can be done in time $O(|V| \cdot |E|)$ [28]. Figure 3 illustrates such decomposition. The decomposition is represented as a tree in which internal vertices corresponds to clique separators and leaves to parts that can not be further decomposed. The subtrees attached to vertex X_i corresponds to the parts of the graph separated by clique X_i. Given such decomposition, it remains to compute the hyperbolicity of the subgraph $G[A_j \cup X_i]$, where A_j is a leaf vertex and X_i is the *father* of A_j in the tree representing the decomposition.

As an example, when the input graph is a $2 \times n$ grid, the pre-processing will decompose the graph using clique-separators of size two (edges) into $n - 1$ 4-cycles. Since 4-cycles are 1-hyperbolic, we conclude directly that the $2 \times n$ grid also is.

In view of the interest of the decomposition by clique-separators, one may ask whether separators of larger diameters could also be used to divide the graph into smaller parts when lower bounds on the hyperbolicity are known. Unfortunately, the problem of finding a $s - t$ separator

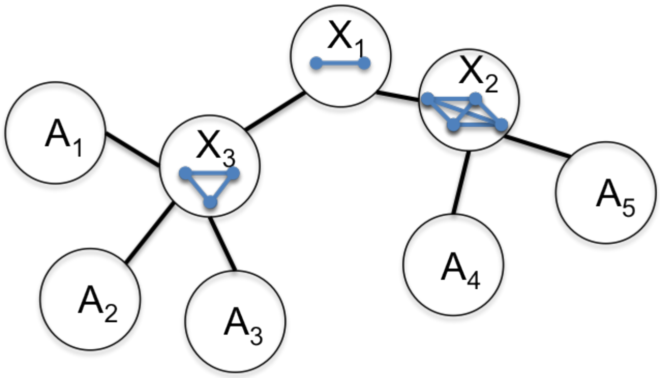


Figure 3: Decomposition by clique-separators.

(i.e., a set X of vertices such that s and t belong to distinct connected components of $G \setminus X$) of diameter h with $h \geq 2$ is NP-hard [19] (more precisely, it is W[1]-hard).

Let us now highlights the interest of a decomposition by clique-separators on some maps of the autonomous system (AS) of the Internet collected by CAIDA and DIMES. We have reported in Table 1 for each map the size of the largest biconnected component (LBCC) and the size of the connected component resulting from the following process: we remove successively from the graph a vertex u if the induced subgraph of its neighbors, $G[\Gamma(u)]$, is a clique (i.e., if the vertex is *simplicial*). This is a restricted usage of a decomposition by clique-separators but it already shows that we can significantly reduce the size of the input graph.

AS map name		**LBCC**	**Removed vertices**	**Final size**
CAIDA	`2010/01/20`	20 940	5 391	15 549
	`2011/01/16`	23 214	6 184	16 950
	`2012/06/01`	25 815	6 747	19 068
DIMES	`12/2010`	18 764	8 056	10 708
	`10/2011`	17 137	7 427	9 710
	`4/2012`	16 907	7 112	9 795

Table 1: Core of some CAIDA and DIMES AS maps, all with hyperbolicity 2.

5 Experimental performances

The algorithm presented in this paper will soon be included (patch #13808 [1]) into Sage, an free mathematics software [3] licensed under the GPL combining the power of many existing open-source packages into a common Python-based interface. Is has been also developed into Grph [2], a Java based library for graph computation.

We have used our implementation in Sage to evaluate the performances of Algorithm 2 coupled with the elimination of simplicial vertices as described in Section 4. We first report some computation results on $N * M$ grids. Next, in Section 5.2 we report experiments on Barabasi-Albert random graphs. Our results show that our algorithm outperforms previous algorithms by orders of magnitude. We then analyze further in Section 5.3 the performances of the algorithm on CAIDA AS maps. In particular, Algorithm 2 allows us to compute the hyperbolicity of the last CAIDA AS maps ($n = 25815$ for the largest biconnected component of the CAIDA map of September 2012) in 8 days while the running time of the algorithm proposed in [17] was estimated to be around 24 days (assuming that 10^{10} operations are performed per second).

5.1 $N \times M$ grids

We have reported in Table 2 the evolution of the computation time of Algorithm 2 on $N \times M$ grids such that $N \times M = 2^2 * 3^2 * 4^2 = 576$ and $N \times M = 2^2 * 3^2 * 5^2 = 900$. The running times are averages over 10 executions of the algorithm. The hyperbolicity of a $N \times M$ grid is $\delta_{N\times M} = \min\{N, M\} - 1$. As expected, the computation time on square grids is way smaller than for grids with sides of very different sizes.

In Section 3, we said that rectangular grids are some of the worst case instances for Algorithm 2. To verify this claim, we have plotted in Figure 4 the relative number of visited 4-tuples with Algorithm 2 compared to the total number of 4-tuples in the graph. Recall that a 4-tuple may be visited up to three times by the algorithm. In these plots, we observe that as soon as the sides of the grids differ by a factor at least 6, then the computation time of Algorithm 2 is larger than the naive algorithm, but it is much faster when the sides of the grid differ by a factor less than 6.

N	M	δ	time
24	24	23	0.08
18	32	17	0.16
16	36	15	1.13
12	48	11	15.01
9	64	8	40.33
8	72	7	50.79
6	96	5	71.80
4	144	3	90.60
3	192	2	97.08
2	288	1	101.82

(a) $N \times M = 576$

N	M	δ	time
30	30	29	0.16
25	36	24	0.18
20	45	19	5.98
18	50	17	23.10
15	60	14	86.94
12	75	11	201.71
10	90	9	297.54
9	100	8	349.51
6	150	5	496.64
5	180	4	537.42
4	225	3	574.37
3	300	2	599.19
2	450	1	622.15

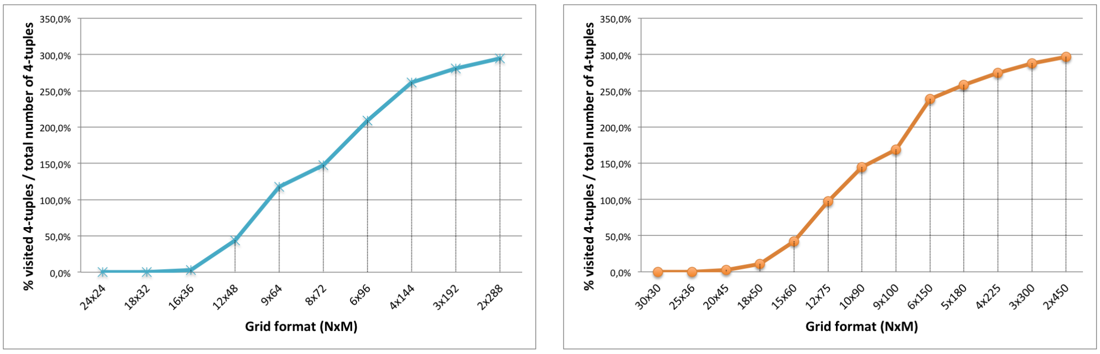
(b) $N \times M = 900$

Table 2: Computation time in secondes of the hyperbolicity of $N \times M$ grids such that $N \times M = 2^2 * 3^2 * 4^2 = 576$ (Table 2a) and $N \times M = 2^2 * 3^2 * 5^2 = 900$ (Table 2b).

5.2 Barabasi-Albert (BA) graphs

Recall that BA random graph model has been proposed in [4] to generate random scale-free graphs. It proceeds as follows: we start with an initial connected graph on $n_0 \geq 2$ nodes (e.g., a random tree), and then we add new nodes with k links using the preferential attachment principle (i.e., the probability of connecting to an existing node depends on its current degree).

To evaluate the performances of Algorithm 2 on BA graphs, we have generated graphs with number of nodes in the range [1 000..10 000] and degree k of new nodes in the range [2..10].



(a) $N \times M = 576$ (b) $N \times M = 900$

Figure 4: Relative performance of Algorithm 2 *vs.* the naive algorithm on $N \times M$ grids.

Reported values are averages over 100 graphs.

We first report in Figure 5 the average hyperbolicity of generated graphs. We observe that the hyperbolicity of BA graphs decreases with the increase of the degree k of newly added nodes, and that for fixed value of k it increases with the number of nodes. These results are explained by the preferential attachment principle which refrains the creation of long induced cycles. The same behavior is observed when changing the size n_0 of the initial graph with some shifting.

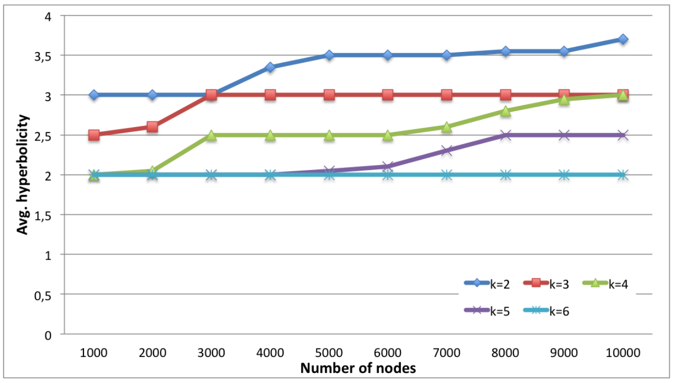
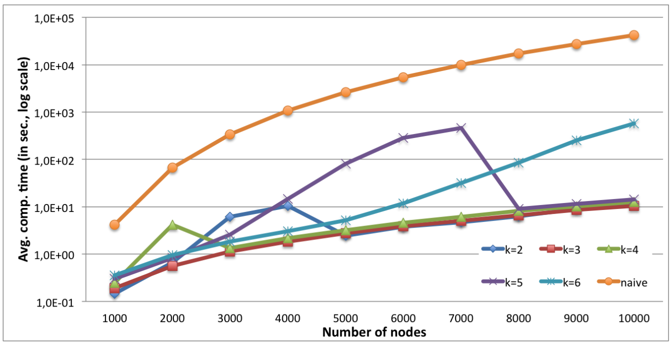


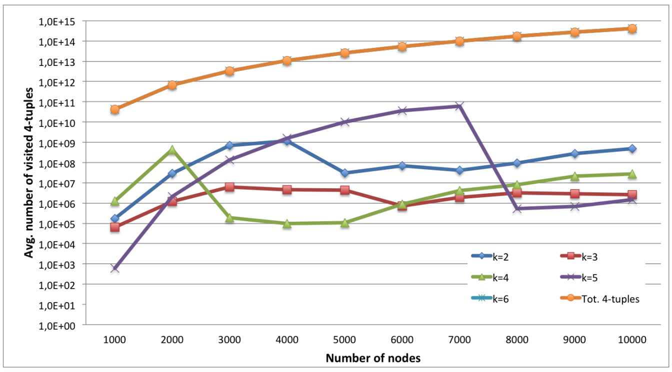
Figure 5: Average hyperbolicity for Barabasi-Albert graphs.

Next, we report in Figure 6b the average number of visited 4-tuples during the execution of the algorithm. The ratio of the average number of visited 4-tuples over the total number of 4-tuples of the graphs varies between 10^{-3} for $N = 7\,000$ and $k = 5$ to 10^{-8} when $N = 10\,000$ and $k = 5$. This highlights the drastic running time improvement of Algorithm 2 over the algorithm proposed in [17] on BA graphs which is from 10^3 to 10^8 times faster. We have to mention that BA graphs have no or very few simplicial vertices and so the decomposition method reported in Section 4 is not helpful in this case.

In Figure 6a we report the average computation time. The general tendency is a slow increase of the computation time with the increase of the parameter k, reflecting that the computation time is in general dominated by the computation of the distances. As reported in Figure 6b, the number of visited 4-tuples is quite small. We also observe a significant increase of the computation time for the particular values of $k = 5$ and $k = 6$.



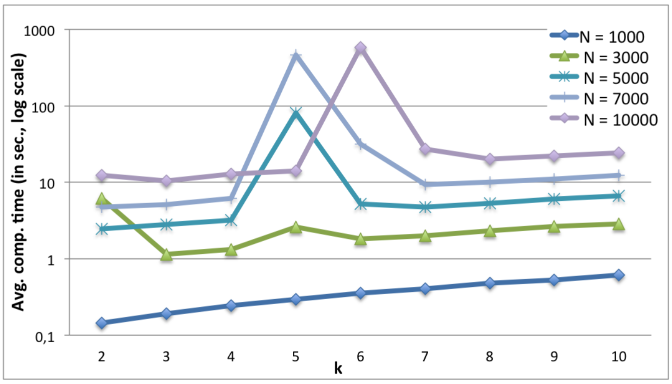
(a) Computation time for Barabasi-Albert graphs.



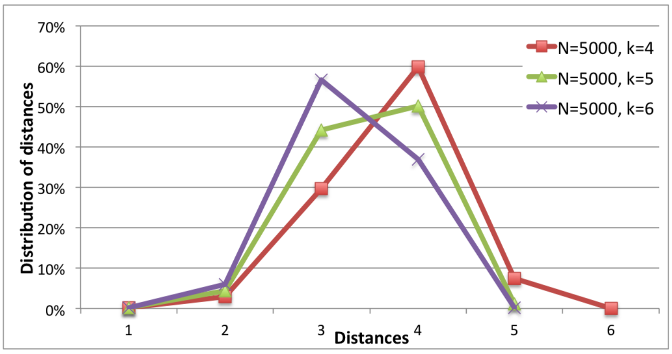
(b) Visited 4-tuples for Barabasi-Albert graphs.

Figure 6: Computation time of the hyperbolicity and visited 4-tuples of Barabasi-Albert graphs.

To better illustrate the impact of the parameter k on the computation time, we plotted in Figure 7a the average computation time function of the parameter k together with the distance distribution in Figure 7b for $k = 4$ to $k = 6$ when $N = 5\,000$. In Figure 7a, we observe peaks on the computation time for particular combinations of the number of nodes and the parameter k, for instance when $N = 5\,000$ and $k = 5$. In order to explain this behavior, recall that the search space is cut each time the algorithm discovers a new lower bound on the hyperbolicity, thus reducing the number of 4-tuple on which it has to iterate. However, if the number of pair of vertices at distance $l \geq 2\delta$ is high, then our algorithm will have to spend time on these pairs of vertices to prove the hyperbolicity. When taking a closer look at the value of the hyperbolicity in Figure 5 and the distance distribution in Figure 7b (for BA graphs with $N = 5\,000$, $k = 5$ and $k = 6$), we remark that the average hyperbolicity is 2, while when $k = 5$ there is a larger proportion of pairs at distance larger than $2\delta = 4$ than with parameter $k = 6$, leading to a higher number of 4-tuples tested. Finally, when $k = 4$, the average hyperbolicity is 2.5 and the proportion of pair of vertices at distance $2\delta = 5$ is very low compared to the graphs generated with $k = 5$ and $k = 6$, leading to a smaller average computation time.



(a) Computation time for Barabasi-Albert graphs.



(b) Distribution of distances in Barabasi-Albert graphs.

Figure 7: Computation time of the hyperbolicity and distance distribution of Barabasi-Albert graphs.

5.3 CAIDA AS maps

We have reported in Table 3 the size of the largest biconnected component (LBCC), the hyperbolicity, the running time and the number of visited 4-tuples during the execution of Algorithm 2 (using decomposition methods presented in Section 4) when computing the hyperbolicity of some CAIDA AS maps. We have selected the maps with different values of the hyperbolicity to highlight the running time improvement of Algorithm 2. As expected with the time complexity expressed in Proposition 6, the running time improvement is correlated to the hyperbolicity of the CAIDA maps. Indeed, Algorithm 2 is more than 100 times faster than the naive algorithm for these maps. Nevertheless, the running times remains large. As explained in Section 3, at each step of the execution of Algorithm 2 we get proven lower and upper bounds for the hyperbolicity of the considered graph. We plotted in Figure 8a the time at which new lower and upper bounds are obtained when computing the hyperbolicity of the maps of Table 3 and in Figure 8b the corresponding number of visited 4-tuples. We observe that a few minutes of computations is sufficient for the lower bound to reach the optimal value but that the time to decrease the upper bounds to the optimal value could be very long. For maps with low hyperbolicity, almost all computation time is spent to prove the optimality of the lower bound.

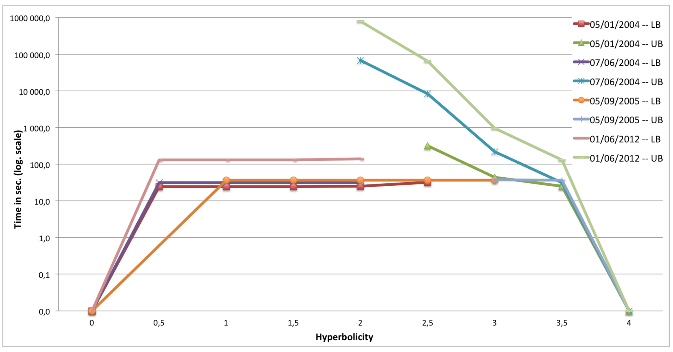
AS map name	**LBCC**	δ	**Time**	**Visited 4-tuples**	**Tot. 4-tuples**
2004/01/05	10 424	2.5	336s	$3.8 \cdot 10^{10}$	$\simeq 4.9 \cdot 10^{14}$
2004/06/07	11 100	2.0	18h50m	$8.8 \cdot 10^{12}$	$\simeq 6.3 \cdot 10^{14}$
2005/09/05	12 957	3.0	58s	$1.2 \cdot 10^{8}$	$\simeq 1.2 \cdot 10^{15}$
2012/06/01	25 815	2.0	8 days	$1.0 \cdot 10^{14}$	$\simeq 1.9 \cdot 10^{16}$

Table 3: Computation time of the hyperbolicity of some CAIDA AS maps.

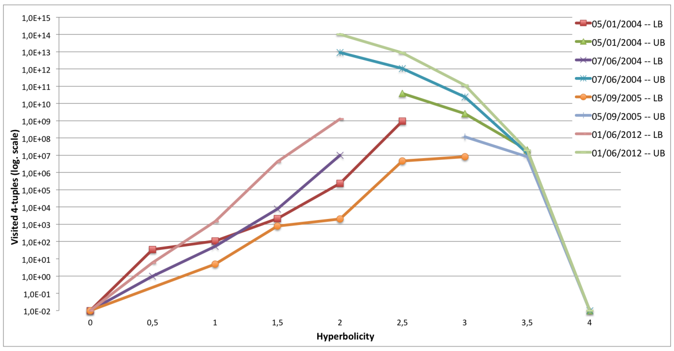
In fact, the computational complexity of Algorithm 2 given in Proposition 6 is in the worst case. It is assumed that the certificate for the hyperbolicity of the input graph G is found with the last visited 4-tuple. It is however likely that this certificate will be found soon in the computations as observed in Figure 8a, increasing earlier the lower bound and reducing the search space. This was observed in all our computations. We tried relabeling randomly the vertices to change the ordering in which pairs of vertices at distance ℓ are considered. This experiment had no impact on the total number of visited 4-tuples. This shows again that certificates for the hyperbolicity are found very early and that the remaining computation time is devoted reducing the upper bound, hence proving the optimality.

6 Hyperbolicity of large-scale graphs

Thanks to Algorithm 2 and to the decomposition method reported in Section 4, we have been able to compute the hyperbolicity of all available CAIDA AS maps since 2004 until June 2012 (190 maps). Although the new algorithm improves upon the algorithm proposed in [17], the overall computation time represents several weeks of computations on a single PC. Indeed, computing the hyperbolicity of the 2012 CAIDA AS map only took 8 days of computations (but maps with hyperbolicity 3 require only a few minutes of computations). The computed values are reported in Figure 9 where we have one dot per CAIDA AS map. The first axis corresponds to the dates the maps where produced. We also reported the linear interpolation of the hyperbolicity.



(a) Time to reach lower and upper bounds of the hyperbolicity of some CAIDA AS maps since 2004. Computation times for lower bounds are plotted from left to right, and from right to left for upper bounds.



(b) Number of visited 4-tuples to reach lower and upper bounds of the hyperbolicity of some CAIDA AS maps since 2004. Number of 4-tuples for lower bounds are plotted from left to right, and from right to left for upper bounds.

Figure 8: Computation time of the hyperbolicity and number of visited 4-tuples of CAIDA AS maps

First, in Figure 9 we observe some measurement bias. For instance, the map of 07/06/2004 has hyperbolicity 2 while maps produced in the months before and after have hyperbolicity 2.5. The same holds for the maps with hyperbolicity 3 (05/09/2005 and 06/02/2006). A frequent variation is also observed between consecutive maps in the period from 2007 till 2009. We still do not know whether this behavior is due to some bias of the measurement or whether it comes from some hidden fact.

The main observation resulting from this experiment is that the hyperbolicity of CAIDA maps has decreased in average from 2.5 to 2 and the hyperbolicity is stable since October 2009. This is not surprising and it is clearly due to the fact that the AS network has become bigger during the last decade. In particular, many new links have appeared and have broken large cycles that made 4-tuples with bigger hyperbolicity.

We have also computed the hyperbolicity of all DIMES AS maps [27] since 2007 (62 maps). These maps are smaller than CAIDA AS maps and so computations are faster. Moreover, around 40% of these nodes are discarded by recursive deletion of simplicial vertices (see Section 4) so that large portions of these maps are quickly eliminated. The results including certificates are

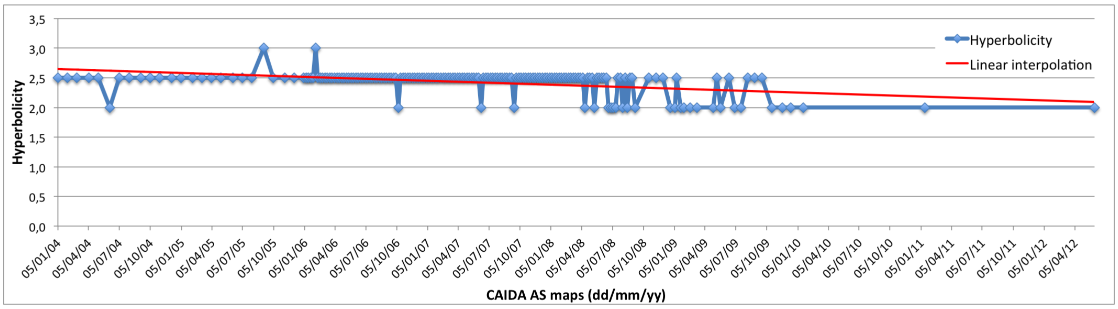


Figure 9: Evolution of the hyperbolicity of CAIDA AS maps since 2004.

reported in Appendix A. We found that all these maps have hyperbolicity 2 but comparison of the running times of Algorithm 2 highlight the structural differences of these maps.

7 Conclusion

We proposed a new exact algorithm to compute the hyperbolicity on large graphs. This algorithm cuts the search space each time a new lower bound is found. We analyzed the complexity of the algorithm and showed the dependency with the size of the largest biconnected component, the computed value of the hyperbolicity and the shortest-path distance distribution. This algorithm can be used as an approximation algorithms. If the complexity of the algorithms remains in $O(n^4)$, we reduce the computational time from an estimated 24 days of computation for the algorithm proposed in [17] (assuming that 10^{10} operations are performed per second) to only 8 days for our algorithm. Thus, we computed the exact value of the hyperbolicity for all the 190 available CAIDA maps and analyzed their evolution. The hyperbolicity of the AS MAPS is slowly decreasing from 2.5 to 2. Some exceptional graphs with hyperbolicity 3 appear from time to time, which can be explained by measurement bias correlated with a smallest number of nodes and edges comparatively to the others maps. We also explained how to use a decomposition by clique-separators to speedup the resolution. We have in particular observed that such decomposition allows to remove around 16% of the vertices of the largest biconnected component of the CAIDA map of 2012 and around 40% of the vertices of DIMES AS maps.

The next step is to implement a parallel version of our algorithms in order to address larger topologies such as the router map of the Internet. Nonetheless, the computation time may still be prohibitive on such large graphs and so new efficient decomposition methods as well as lower-bound heuristics would be helpful. We shall also further analyze the presence of a large number of simplicial vertices in AS maps.

Acknowledgments

This work has been partially supported FP7 STREP EULER, and ANRs AGAPE and GRATEL.

References

[1] Gromov hyperbolicity of graphs. http://trac.sagemath.org/sage_trac/ticket/13808.

[2] Grph. http://www-sop.inria.fr/mascotte/software/grph/, 2012.

[3] Sage open-source mathematical software system, version 5.4.1. http://www.sagemath.org, 2012.

[4] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.

[5] H.-J. Bandelt and V. Chepoi. 1-hyperbolic graphs. *SIAM J. Discrete Math.*, 16(2):323–334, 2003.

[6] M. Boguna, F. Papadopoulos, and D. Krioukov. Sustaining the Internet with Hyperbolic Mapping. *Nature Communications*, 1(62), Oct 2010.

[7] G. Brinkmann, J. H. Koolen, and V. Moulton. On the hyperbolicity of chordal graphs. *Annals of Combinatorics*, 2001.

[8] W. Carballosa, J.M. Rodríguez, J. M. Sigarreta, and M. Villeta. Gromov hyperbolicity of line graphs. *The Electronic Journal of Combinatorics*, 18(P210):1–18, 2011.

[9] W. Chen, W. Fang, G. Hu, and M. W. Mahoney. On the hyperbolicity of small-world and tree-like random graphs. Technical Report arXiv:1201.1717, ArXiv, September 2012.

[10] V. Chepoi, F. F. Dragan, B. Estellon, M. Habib, and Y. Vaxès. Notes on diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs. *Electronic Notes in Discrete Mathematics*, 31:231–234, 2008.

[11] CRAN: The Comprehensive R Archive Network project. http://cran.r-project.org/.

[12] F. de Montgolfier, M. Soto, and L. Viennot. Treewidth and hyperbolicity of the internet. In *10th IEEE International Symposium on Networking Computing and Applications (NCA)*, pages 25–32, Cambridge, Massachusetts, USA, August 2011. IEEE Computer Society.

[13] distory: Distance between phylogenetic histories. http://cran.r-project.org/web/packages/distory/index.html.

[14] A. Dress, K. Huber, J. Koolen, V. Moulton, and A. Spillner. *Basic Phylogenetic Combinatorics*. Cambridge University Press, 2012.

[15] R. Duan and S. Pettie. Fast algorithms for (max, min)-matrix multiplication and bottleneck shortest paths. In Claire Mathieu, editor, *SODA*, pages 384–391, New York, NY, USA, January 2009. SIAM.

[16] The Cooperative Association for Internet Data Analysis (CAIDA). The caida as relationships dataset. http://www.caida.org/data/active/as-relationships/.

[17] H. Fournier, A. Ismail, and A. Vigneron. Computing the Gromov hyperbolicity of a discrete metric space. Technical Report arXiv:1210.3323, ArXiv, October 2012.

[18] M. Gromov. Hyperbolic groups. *Essays in Group Theory*, 8:75–263, 1987.

[19] Pinar Heggernes, Pim van 't Hof, Dániel Marx, Neeldhara Misra, and Yngve Villanger. On the parameterized complexity of finding separators with non-hereditary properties. In *WG*, pages 332–343, 2012.

[20] Edmond Jonckheere and Poonsuk Lohsoonthorn. A hyperbolic geometric approach to multipath routing. *Conference on Control and Automation*, 2002.

[21] Edmond Jonckheere and Poonsuk Lohsoonthorn. Geometry of network security. *American Control Conference*, 2, 2004.

[22] D. Knuth. *The art of computer programming*, volume 4, Fascicle 2, chapter Generating All tuples and permutations. Addison-Wesley, February 2005.

[23] J. H. Koolen and V. Moulton. Hyperbolic bridged graphs. *Eur. J. Comb.*, 23(6):683–699, 2002.

[24] O. Narayan and I. Saniee. The Large Scale Curvature of Networks. *ArXiv e-prints*, July 2009.

[25] F. Papadopoulos, D. Krioukov, M. Boguna, and A. Vahdat. Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces. *SIGMETRICS Performance Evaluation Review*, 37(2):15–17, 2009.

[26] J.M. Rodríguez and J. M. Sigarreta. Bounds on Gromov hyperbolicity constant in graphs. In *Proceedings Indian Acad. Sci. (Mathematical Sciences)*, volume 122, pages 53–65, 2012.

[27] Y. Shavitt and E. Shir. DIMES: Let the Internet Measure Itself. *CoRR*, abs/cs/0506099, 2005.

[28] Robert E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221 – 232, 1985.

[29] Y. Wu and C. Zhang. Hyperbolicity and chordality of a graph. *Electr. J. Comb.*, 18(1), 2011.

A Simple results

Lemma 10. *Cycles of order $n = 4p + \varepsilon$, with $p \geq 1$ and $\varepsilon \in \{0, 1, 2, 3\}$, are $(p - 1/2)$-hyperbolic when $\varepsilon = 1$, and p-hyperbolic otherwise.*

Proof of Lemma 10. Let C be a cycle of order $n = 4p+\varepsilon$, with $p \geq 1$ and $\varepsilon \in \{0, 1, 2, 3\}$, and let the vertices be indexed $\{0, 1, \cdots, 4p + \varepsilon - 1\}$. Let a, b, c, d be four vertices of C. In order to break symmetries and w.l.o.g. we assume that $a = 0 < b \leq p < c \leq 2p+\lfloor\varepsilon/2\rfloor < d \leq 3p+\lfloor\varepsilon/2\rfloor+\lfloor\varepsilon/3\rfloor$, and so we have

$$1 \leq \mathbb{d}(a, b) \leq p \tag{1a}$$
$$p + 1 \leq \mathbb{d}(a, c) \leq 2p + \lfloor\varepsilon/2\rfloor \tag{1b}$$
$$p + \varepsilon - \lfloor\varepsilon/2\rfloor - \lfloor\varepsilon/3\rfloor \leq \mathbb{d}(a, d) \leq 2p + \varepsilon - \lfloor\varepsilon/2\rfloor - 1 \tag{1c}$$
$$1 \leq \mathbb{d}(b, c) \leq 2p + \lfloor\varepsilon/2\rfloor - 1 \tag{1d}$$
$$p + \lfloor\varepsilon/2\rfloor + 1 \leq \mathbb{d}(b, d) \leq 2p + \lfloor\varepsilon/2\rfloor \tag{1e}$$
$$1 \leq \mathbb{d}(c, d) \leq 2p + \lfloor\varepsilon/2\rfloor + \lfloor\varepsilon/3\rfloor - 1 \tag{1f}$$

and,

$$2 \leq S_1 = \mathbb{d}(a, b) + \mathbb{d}(c, d) \leq 3p + \lfloor\varepsilon/2\rfloor + \lfloor\varepsilon/3\rfloor - 1 \tag{1g}$$
$$2p + \lfloor\varepsilon/2\rfloor + 2 \leq S_2 = \mathbb{d}(a, c) + \mathbb{d}(b, d) \leq 4p + 2\lfloor\varepsilon/2\rfloor \tag{1h}$$
$$p + \varepsilon - \lfloor\varepsilon/2\rfloor - \lfloor\varepsilon/3\rfloor + 1 \leq S_3 = \mathbb{d}(a, d) + \mathbb{d}(b, c) \leq 4p + \varepsilon - 2 \tag{1i}$$

For any values of b, c, d, we have $S_1 + S_3 = \mathbb{d}(a, b) + \mathbb{d}(c, d) + \mathbb{d}(a, d) + \mathbb{d}(b, c) = 4p + \varepsilon$, and so maximizing S_1 decreases S_3 and conversely.

Observe that the maximum value of S_1 is obtained with $b = p$, $c = p + 1$, and $d = 3p + \lfloor\varepsilon/2\rfloor + \lfloor\varepsilon/3\rfloor$. For these values we have $S_2 = 3p + \lfloor\varepsilon/2\rfloor + \lfloor\varepsilon/3\rfloor + 1 > S_1$, $S_3 = p + \varepsilon - \lfloor\varepsilon/2\rfloor - \lfloor\varepsilon/3\rfloor + 1 < S_1$, and so $h(a, b, c, d) = 1$. Similarly, the maximum value of S_3 is obtained with $b = 1$, $c = 2p + \lfloor\varepsilon/2\rfloor$, and $d = 2p + \lfloor\varepsilon/2\rfloor + 1$, which gives $S_1 = 2$, $S_2 = 4p + 2\lfloor\varepsilon/2\rfloor$, and so $h(a, b, c, d) = |\varepsilon - 2 - 2\lfloor\varepsilon/2\rfloor| \leq 2$.

Let us now assume that $S_2 = \max\{S_1, S_2, S_3\}$. Since $S_1 + S_3 = 4p + \varepsilon$, the best we can do to maximize $h(a, b, c, d)$ is to choose values of a, b, c, d such that the maximum among S_1 and S_3 is the closest possible to $2p + \varepsilon/2$, and such that S_2 is as large as possible. This can be achieve choosing $b = p$, $c = 2p + \lfloor\varepsilon/2\rfloor$, and $d = 3p + \lfloor\varepsilon/2\rfloor + \lfloor\varepsilon/3\rfloor$. Indeed, with these values, we get $S_2 = 4p+2\lfloor\varepsilon/2\rfloor$, so the maximum possible value, and $S_1 = 2p+\lfloor\varepsilon/3\rfloor$ and $S_3 = 2p+\varepsilon-\lfloor\varepsilon/3\rfloor \geq S_1$ which are the closest possible of $2p + \varepsilon/2$. We obtain $h(a, b, c, d) = 2p + 2\lfloor\varepsilon/2\rfloor - \varepsilon + \lfloor\varepsilon/3\rfloor$.

Since $\varepsilon \in \{0, 1, 2, 3\}$, it follows that cycles of order $n = 4p + \varepsilon$, with $p \geq 1$, are $(p - 1/2)$-hyperbolic when $\varepsilon = 1$, and p-hyperbolic otherwise. □

Lemma 11. *$n \times m$ grids, with $2 \leq n \leq m$, are $(n - 1)$-hyperbolic.*

Proof of Lemma 11. Let $\{v_{i,j},\ 1 \leq i \leq n,\ 1 \leq j \leq m\}$ the vertices of the $n \times m$ grid.

Firstly, suppose that $n = m$, and consider the four vertices $v_{1,1}, v_{1,n}, v_{n,1}, v_{n,n}$. We have $S_1 = \mathbb{d}(v_{1,1}, v_{1,n}) + \mathbb{d}(v_{n,1}, v_{n,n}) = 2n - 2$, $S_2 = \mathbb{d}(v_{1,1}, v_{n,n}) + \mathbb{d}(v_{1,n}, v_{n,1}) = 4n - 4$, $S_3 = \mathbb{d}(v_{1,1}, v_{n,1}) + \mathbb{d}(v_{1,n}, v_{n,n}) = 2n - 2$, and so $h(v_{1,1}, v_{1,n}, v_{n,1}, v_{n,n}) = 2n - 2$. Since $h(a, b, c, d)$ is less or equal to the diameter of the graph for any 4-tuple (a, b, c, d), this value is optimal.

Secondly, let $n \leq x \leq m$ and consider the four vertices $v_{1,1}, v_{1,x}, v_{n,1}, v_{n,x}$. We have $S_1 = \mathbb{d}(v_{1,1}, v_{1,x}) + \mathbb{d}(v_{n,1}, v_{n,x}) = 2x - 2$, $S_2 = \mathbb{d}(v_{1,1}, v_{n,x}) + \mathbb{d}(v_{1,x}, v_{n,1}) = 2n + 2x - 4$, $S_3 = \mathbb{d}(v_{1,1}, v_{n,1}) + \mathbb{d}(v_{1,x}, v_{n,x}) = 2n - 2$, and since $S_2 \geq S_1 \geq S_3$ we obtain $h(v_{1,1}, v_{1,x}, v_{n,1}, v_{n,x}) = 2n - 2$.

Let now $v_{i_1,j_1}, v_{i_2,j_2}, v_{i_3,j_3}, v_{i_4,j_4}$ be four vertices of the grid such that $1 < i_1, i_2 \leq n/2 < i_3, i_4 < n$, and $1 < j_1, j_3 \leq n/2 < j_2, j_4 < n$, and let us show that $h(v_{i_1,j_1}, v_{i_2,j_2}, v_{i_3,j_3}, v_{i_4,j_4}) \leq h(v_{i_1-1,j_1-1}, v_{i_2,j_2}, v_{i_3,j_3}, v_{i_4,j_4})$. We have $S_1 = \mathbb{d}(v_{i_1,j_1}, v_{i_2,j_2}) + \mathbb{d}(v_{i_3,j_3}, v_{i_4,j_4})$, $S_2 = \mathbb{d}(v_{i_1,j_1}, v_{i_3,j_3}) + \mathbb{d}(v_{i_2,j_2}, v_{i_4,j_4})$, and $S_3 = \mathbb{d}(v_{i_1,j_1}, v_{i_4,j_4}) + \mathbb{d}(v_{i_2,j_2}, v_{i_3,j_3})$.

According the shortest path distance in the grid, we have

$$\mathbb{d}(v_{i_1,j_1}, v_{i_2,j_2}) + 1 \leq \mathbb{d}(v_{i_1-1,j_1-1}, v_{i_2,j_2}) \leq \mathbb{d}(v_{i_1,j_1}, v_{i_2,j_2}) + 2 \tag{2a}$$
$$\mathbb{d}(v_{i_1,j_1}, v_{i_3,j_3}) + 1 \leq \mathbb{d}(v_{i_1-1,j_1-1}, v_{i_3,j_3}) \leq \mathbb{d}(v_{i_1,j_1}, v_{i_3,j_3}) + 2 \tag{2b}$$
$$\mathbb{d}(v_{i_1-1,j_1-1}, v_{i_4,j_4}) = \mathbb{d}(v_{i_1,j_1}, v_{i_4,j_4}) + 2 \tag{2c}$$

and so

$$S_1 + 1 \leq S'_1 = \mathbb{d}(v_{i_1-1,j_1-1}, v_{i_2,j_2}) + \mathbb{d}(v_{i_3,j_3}, v_{i_4,j_4}) \leq S_1 + 2 \tag{2d}$$
$$S_2 + 1 \leq S'_2 = \mathbb{d}(v_{i_1-1,j_1-1}, v_{i_3,j_3}) + \mathbb{d}(v_{i_2,j_2}, v_{i_4,j_4}) \leq S_2 + 2 \tag{2e}$$
$$S'_3 = \mathbb{d}(v_{i_1-1,j_1-1}, v_{i_4,j_4}) + \mathbb{d}(v_{i_2,j_2}, v_{i_3,j_3}) = S_3 + 2 \tag{2f}$$

From these values, we get $|S'_1 - S'_2| \leq |S_1 - S_2| + 1$, $|S'_1 - S'_3| \leq |S_1 - S_3| + 1$, and $|S'_2 - S'_3| \leq |S_2 - S_3| + 1$. Therefore, we have $h(v_{i_1,j_1}, v_{i_2,j_2}, v_{i_3,j_3}, v_{i_4,j_4}) \leq h(v_{i_1-1,j_1-1}, v_{i_2,j_2}, v_{i_3,j_3}, v_{i_4,j_4})$, and we can prove the same result when decreasing some of the values of i_1, i_2, j_1, j_3 or increasing some of the values of i_3, i_4, j_2, j_4.

Altogether, we conclude that $n \times m$ grids, with $2 \leq n \leq m$, are $(n-1)$-hyperbolic. □

Lemma 12. *The d-dimensional grid of side s is* $(s-1)\lfloor d/2 \rfloor$*-hyperbolic.*

Proof. The vertices of the d-dimensional grid of side s are labeled $(u_0, u_1, \cdots, u_{d-1})$, with $u_i \in [0..s-1]$ for $i \in [0..d-1]$, and there is an edge between $u = (u_0, u_1, \cdots, u_{d-1})$ and $v = (v_0, v_1, \cdots, v_{d-1})$ if $u_i = v_i$ for $i \in [0..j-1] \cup [j+1..d-1]$ and either $v_j = u_j + 1$ if $u_j + 1 < s$ or $v_j = u_j - 1$ if $u_j - 1 \geq 0$.

Consider now the four vertices u, v, w, x such that

$$u = (0, 0, \cdots, 0) \quad \text{contains only 0's} \tag{3a}$$
$$v = (s-1, s-1, \cdots, s-1) \quad \text{contains only } s-1\text{'s} \tag{3b}$$
$$w = (w_0, w_1, \cdots, w_{d-1}) \quad \text{contains } \lfloor d/2 \rfloor \text{ 0's and } \lceil d/2 \rceil\ s-1\text{'s} \tag{3c}$$
$$x = (x_0, x_1, \cdots, x_{d-1}) \quad \text{obtained from } w \text{ replacing each 0 with a } s-1 \text{ and each } s-1 \text{ with a 0} \tag{3d}$$

We have for instance $\mathbb{d}(u, w) = (s-1)\lceil d/2 \rceil$, and so

$$S_1 = \mathbb{d}(u, v) + \mathbb{d}(w, x) = 2(s-1)d \tag{3e}$$
$$S_2 = \mathbb{d}(u, w) + \mathbb{d}(v, x) = 2(s-1)\lceil d/2 \rceil \tag{3f}$$
$$S_3 = \mathbb{d}(u, x) + \mathbb{d}(v, w) = 2(s-1)\lfloor d/2 \rfloor \tag{3g}$$

We obtain $h(u, v, w, x) = S_1 - S_3 = 2(s-1)\lfloor d/2 \rfloor$.

The optimality proof is similar to the proof of Lemma 11. □

Corollary 13. *The hypercube* $\mathcal{H}_2(n)$ *of dimension n and order* 2^n *is* $\lfloor n/2 \rfloor$*-hyperbolic.*

Proof of Corollary 13. The proof follows from Lemma 12 since the n-dimensional grid of side 2 is the hypercube $\mathcal{H}_2(n)$. □

B Hyperbolicity of some CAIDA AS maps

Table 4 lists the hyperbolicity of all CAIDA AS maps since 2004 [16]. We have also reported in this table the number of 4-tuples with given hyperbolicity among a large set of randomly chosen 4-tuples. We can derive from these values the statistical distribution of the hyperbolicity of the 4-tuples. Last, we have reported some hyperbolicity *certificates*, i.e., a 4-tuple with maximum hyperbolicity.

Table 4: Hyperbolicity of CAIDA AS maps since 2004.

AS map name	δ	Certificate	Visited 4-tuples	Total number
2004/01/05	5/2	3233, 8338, 8923, 13268	3.9 10^{10}	4.9 10^{14}
2004/02/02	5/2	12497, 20807, 28983, 29226	2.4 10^{10}	5.1 10^{14}
2004/03/01	5/2	8863, 9249, 20515, 28776	3.1 10^{10}	5.3 10^{14}
2004/04/05	5/2	12316, 13092, 21172, 28949	1.6 10^{11}	5.9 10^{14}
2004/05/03	5/2	5434, 12660, 12764, 20633	4.2 10^{10}	6.2 10^{14}
2004/06/07	2	6802, 25454, 30838, 31318	8.8 10^{12}	6.3 10^{14}
2004/07/05	5/2	1955, 6802, 20685, 25454	7.7 10^{10}	6.6 10^{14}
2004/08/02	5/2	2148, 12644, 13105, 29335	5.4 10^{10}	7.0 10^{14}
2004/09/06	5/2	6802, 12615, 16010, 20816	3.7 10^{10}	7.3 10^{14}
2004/10/04	5/2	5434, 12550, 12644, 21341	5.4 10^{10}	7.7 10^{14}
2004/11/01	5/2	5434, 13042, 21341, 30768	6.4 10^{10}	8.0 10^{14}
2004/12/06	5/2	5379, 25454, 25496, 31340	5.5 10^{10}	8.4 10^{14}
2005/01/03	5/2	8880, 13121, 28903, 34214	8.8 10^{10}	8.10 10^{14}
2005/02/07	5/2	6802, 8880, 8923, 25496	9.10 10^{10}	8.10 10^{14}
2005/03/07	5/2	13302, 15378, 15775, 24825	1.2 10^{11}	9.1 10^{14}
2005/04/04	5/2	13105, 28903, 29632, 31258	7.5 10^{10}	9.5 10^{14}
2005/05/02	5/2	6850, 28903, 28949, 29233	8.8 10^{10}	1.0 10^{15}
2005/06/06	5/2	15378, 15775, 24607, 34784	1.1 10^{11}	1.0 10^{15}
2005/07/04	5/2	6736, 8226, 12880, 15595	1.1 10^{11}	1.1 10^{15}
2005/08/01	5/2	8471, 12983, 17480, 21103	1.0 10^{11}	1.1 10^{15}
2005/09/05	3	12739, 12764, 16010, 34645	1.2 10^{8}	1.2 10^{15}
2005/10/03	5/2	8825, 12507, 25306, 31549	6.6 10^{10}	1.2 10^{15}
2005/11/07	5/2	20702, 29390, 29663, 34645	9.7 10^{10}	1.3 10^{15}
2005/12/05	5/2	13365, 16070, 16138, 29442	8.1 10^{10}	1.3 10^{15}
2006/01/02	5/2	13071, 15682, 21011, 29390	1.2 10^{11}	1.5 10^{15}
2006/01/09	5/2	10747, 16070, 28733, 30952	1.5 10^{11}	1.4 10^{15}
2006/01/16	5/2	24416, 29632, 31258, 33818	1.0 10^{11}	1.4 10^{15}
2006/01/23	5/2	10687, 16070, 16130, 29663	1.3 10^{11}	1.4 10^{15}
2006/01/30	5/2	8825, 12507, 29630, 31589	1.3 10^{11}	1.4 10^{15}
2006/02/06	3	1955, 13210, 20722, 34416	4.3 10^{10}	1.4 10^{15}
2006/02/13	5/2	21411, 28752, 29034, 29107	1.2 10^{11}	1.5 10^{15}
2006/02/20	5/2	20778, 28858, 31077, 34775	1.5 10^{11}	1.5 10^{15}
2006/02/27	5/2	2549, 6458, 10490, 17086	1.1 10^{12}	1.5 10^{15}
2006/03/06	5/2	12764, 21411, 29034, 29107	1.2 10^{11}	1.5 10^{15}
2006/03/13	5/2	16190, 29663, 35032, 35147	1.1 10^{11}	1.4 10^{15}
2006/03/20	5/2	13310, 30861, 31023, 34379	1.8 10^{11}	1.5 10^{15}
2006/03/27	5/2	7202, 8982, 15393, 34771	9.2 10^{10}	1.5 10^{15}

Continued on next page

Table 4 – continued from previous page

AS map name	δ	Certificate	Visited 4-tuples	Total number
2006/04/03	5/2	15614, 20803, 31340, 38932	1.5 10^{11}	1.5 10^{15}
2006/04/10	5/2	15614, 20490, 31340, 38932	9.10 10^{10}	1.5 10^{15}
2006/04/17	5/2	13215, 23554, 28949, 31340	1.8 10^{11}	1.6 10^{15}
2006/04/24	5/2	15614, 20490, 31340, 38932	1.0 10^{11}	1.5 10^{15}
2006/05/01	5/2	12764, 15632, 16010, 16190	1.4 10^{11}	1.5 10^{15}
2006/05/08	5/2	2148, 8745, 29329, 31340	1.3 10^{11}	1.6 10^{15}
2006/05/15	5/2	2148, 23697, 24287, 34495	1.2 10^{11}	1.6 10^{15}
2006/05/22	5/2	2148, 23697, 24287, 34495	7.8 10^{10}	1.5 10^{15}
2006/05/29	5/2	3233, 9087, 12297, 30890	1.9 10^{11}	1.6 10^{15}
2006/06/05	5/2	12677, 19942, 28911, 34238	1.4 10^{12}	1.6 10^{15}
2006/06/12	5/2	5598, 13000, 13310, 31556	1.3 10^{11}	1.6 10^{15}
2006/06/19	5/2	5598, 12742, 21257, 24703	2.0 10^{11}	1.6 10^{15}
2006/06/26	5/2	12690, 18256, 20545, 24997	1.4 10^{11}	1.6 10^{15}
2006/07/03	5/2	12690, 20545, 24005, 24997	1.4 10^{11}	1.7 10^{15}
2006/07/10	5/2	9560, 20722, 35048, 39581	1.2 10^{11}	1.7 10^{15}
2006/07/17	5/2	13210, 34238, 34771, 39031	1.10 10^{11}	1.6 10^{15}
2006/07/24	5/2	5379, 16226, 28949, 33961	1.7 10^{11}	1.7 10^{15}
2006/07/31	5/2	5598, 13310, 30953, 41166	1.4 10^{11}	1.7 10^{15}
2006/08/07	5/2	8389, 12452, 12764, 16010	1.7 10^{11}	1.8 10^{15}
2006/08/14	5/2	12764, 16010, 25306, 29271	1.0 10^{11}	1.7 10^{15}
2006/08/21	5/2	6802, 15536, 29030, 30797	1.4 10^{11}	1.8 10^{15}
2006/08/28	5/2	12764, 15700, 16010, 29271	2.3 10^{11}	1.8 10^{15}
2006/09/04	5/2	5379, 12987, 28949, 39505	9.7 10^{10}	1.8 10^{15}
2006/09/11	5/2	8389, 12452, 12764, 16010	1.10 10^{11}	1.8 10^{15}
2006/09/18	5/2	10037, 12690, 20545, 28751	1.0 10^{11}	1.8 10^{15}
2006/09/25	5/2	12764, 16010, 25306, 29271	1.0 10^{11}	1.9 10^{15}
2006/10/02	5/2	16118, 34163, 34797, 35212	2.1 10^{11}	1.8 10^{15}
2006/10/09	2	14271, 18202, 24618, 38022	2.8 10^{13}	1.8 10^{15}
2006/10/16	5/2	8641, 30952, 34797, 35212	1.6 10^{11}	1.8 10^{15}
2006/10/23	5/2	13310, 21257, 21433, 34331	1.4 10^{11}	1.9 10^{15}
2006/10/30	5/2	21261, 34331, 34787, 38070	1.0 10^{11}	1.8 10^{15}
2006/11/06	5/2	4796, 9326, 9950, 17556	1.1 10^{11}	1.8 10^{15}
2006/11/13	5/2	9303, 15736, 31253, 38067	1.7 10^{11}	1.0 10^{15}
2006/11/20	5/2	20490, 24203, 25349, 30362	1.5 10^{12}	1.10 10^{15}
2006/11/27	5/2	21257, 24688, 29012, 34118	1.0 10^{11}	1.9 10^{15}
2006/12/04	5/2	17808, 17829, 2055, 38022	1.8 10^{11}	1.9 10^{15}
2006/12/11	5/2	12773, 20516, 20699, 39307	1.0 10^{11}	2.0 10^{15}
2006/12/18	5/2	8779, 13307, 20516, 21437	1.9 10^{11}	2.1 10^{15}
2006/12/25	5/2	13121, 25507, 29012, 34867	2.5 10^{11}	2.1 10^{15}
2007/01/01	5/2	9950, 10049, 18356, 24475	1.4 10^{11}	2.1 10^{15}
2007/01/08	5/2	9950, 10049, 18356, 24475	1.4 10^{11}	2.1 10^{15}
2007/01/15	5/2	3836, 4796, 18362, 24198	2.1 10^{11}	2.1 10^{15}
2007/01/22	5/2	1985, 4796, 9791, 9950	1.4 10^{11}	2.1 10^{15}
2007/01/29	5/2	8430, 20633, 39200, 41419	9.2 10^{10}	1.2 10^{15}
2007/02/05	5/2	3836, 4796, 14329, 37977	1.3 10^{11}	2.1 10^{15}
2007/02/12	5/2	1985, 4796, 9791, 9950	1.1 10^{11}	2.1 10^{15}

Continued on next page

Table 4 – continued from previous page

AS map name	δ	Certificate	Visited 4-tuples	Total number
2007/02/19	5/2	16309, 21488, 24618, 35483	$2.9\ 10^{11}$	$2.1\ 10^{15}$
2007/02/26	5/2	15458, 16231, 24618, 41308	$1.7\ 10^{11}$	$2.2\ 10^{15}$
2007/03/05	5/2	4855, 9881, 24490, 38296	$4.9\ 10^{12}$	$2.3\ 10^{15}$
2007/03/12	5/2	1727, 4796, 9791, 9950	$1.4\ 10^{11}$	$2.3\ 10^{15}$
2007/03/19	5/2	1727, 4796, 9791, 9950	$1.4\ 10^{11}$	$2.3\ 10^{15}$
2007/03/26	5/2	12790, 13107, 39528, 41684	$1.5\ 10^{11}$	$2.4\ 10^{15}$
2007/04/02	5/2	28858, 39528, 41390, 41684	$1.2\ 10^{11}$	$2.3\ 10^{15}$
2007/04/09	5/2	1955, 2549, 31218, 41832	$2.0\ 10^{11}$	$2.4\ 10^{15}$
2007/04/16	5/2	18082, 29012, 31347, 34870	$1.5\ 10^{11}$	$2.4\ 10^{15}$
2007/04/23	5/2	18082, 29012, 31347, 34870	$1.1\ 10^{11}$	$2.3\ 10^{15}$
2007/04/30	5/2	12790, 16070, 39528, 41684	$1.2\ 10^{11}$	$2.5\ 10^{15}$
2007/05/07	5/2	15700, 21257, 25386, 31023	$9.2\ 10^{10}$	$2.4\ 10^{15}$
2007/05/14	5/2	9881, 11189, 31253, 38155	$2.8\ 10^{11}$	$2.5\ 10^{15}$
2007/05/21	5/2	21257, 25386, 31023, 35339	$1.1\ 10^{11}$	$2.5\ 10^{15}$
2007/05/28	5/2	12374, 13107, 20699, 34792	$5.10\ 10^{11}$	$2.5\ 10^{15}$
2007/06/04	5/2	18393, 20459, 31253, 38155	$2.6\ 10^{11}$	$2.5\ 10^{15}$
2007/06/11	2	18308, 34411, 35710, 38941	$3.8\ 10^{13}$	$2.6\ 10^{15}$
2007/06/18	5/2	4621, 10202, 24455, 24514	$3.9\ 10^{11}$	$2.5\ 10^{15}$
2007/06/25	5/2	4796, 9326, 23596, 23900	$3.3\ 10^{11}$	$2.6\ 10^{15}$
2007/07/02	5/2	28790, 31598, 33974, 34094	$4.2\ 10^{11}$	$2.6\ 10^{15}$
2007/07/09	5/2	28790, 31598, 33974, 34094	$3.4\ 10^{11}$	$2.7\ 10^{15}$
2007/07/16	5/2	20797, 21061, 28886, 34620	$1.8\ 10^{12}$	$2.6\ 10^{15}$
2007/07/23	5/2	3836, 4796, 23900, 38155	$2.5\ 10^{11}$	$2.6\ 10^{15}$
2007/07/30	5/2	20516, 25507, 33974, 41338	$2.8\ 10^{11}$	$2.7\ 10^{15}$
2007/08/06	5/2	16007, 24609, 29685, 33974	$2.1\ 10^{12}$	$2.7\ 10^{15}$
2007/08/13	5/2	10364, 21257, 33974, 38962	$1.2\ 10^{11}$	$2.7\ 10^{15}$
2007/08/20	5/2	20516, 24681, 33974, 35409	$1.2\ 10^{11}$	$2.6\ 10^{15}$
2007/08/27	5/2	31023, 33974, 34870, 35795	$2.5\ 10^{11}$	$2.6\ 10^{15}$
2007/09/03	5/2	10364, 21257, 33974, 38962	$1.8\ 10^{11}$	$2.7\ 10^{15}$
2007/09/10	5/2	12976, 20294, 20578, 21104	$1.8\ 10^{12}$	$2.7\ 10^{15}$
2007/09/17	2	11025, 32262, 32703, 35183	$9.4\ 10^{9}$	$3.1\ 10^{13}$
2007/09/24	5/2	4796, 9326, 23596, 23900	$2.2\ 10^{11}$	$2.8\ 10^{15}$
2007/10/01	5/2	4796, 9326, 23596, 23900	$2.2\ 10^{11}$	$2.9\ 10^{15}$
2007/10/08	5/2	4855, 9881, 17827, 24490	$1.5\ 10^{12}$	$2.8\ 10^{15}$
2007/10/15	5/2	21104, 21131, 34742, 41588	$1.6\ 10^{12}$	$2.8\ 10^{15}$
2007/10/22	5/2	29275, 42802, 43022, 43120	$1.9\ 10^{11}$	$2.9\ 10^{15}$
2007/10/29	5/2	31489, 33974, 43111, 43120	$1.4\ 10^{11}$	$2.9\ 10^{15}$
2007/11/05	5/2	21497, 31023, 34265, 43435	$1.7\ 10^{11}$	$2.9\ 10^{15}$
2007/11/12	5/2	25119, 31166, 34742, 41588	$2.7\ 10^{12}$	$2.9\ 10^{15}$
2007/11/19	5/2	4855, 9881, 24490, 38296	$3.2\ 10^{12}$	$2.10\ 10^{15}$
2007/11/26	5/2	4855, 9881, 24490, 38296	$3.4\ 10^{12}$	$2.10\ 10^{15}$
2007/12/03	5/2	21497, 31023, 34265, 43435	$1.9\ 10^{11}$	$2.10\ 10^{15}$
2007/12/10	5/2	21497, 31023, 34265, 43435	$1.6\ 10^{11}$	$2.10\ 10^{15}$
2007/12/17	5/2	4855, 9881, 24490, 38296	$2.7\ 10^{12}$	$2.9\ 10^{15}$
2007/12/24	5/2	4855, 9881, 24490, 38296	$3.1\ 10^{12}$	$3.3\ 10^{15}$
2007/12/31	5/2	21497, 31023, 34265, 43435	$1.5\ 10^{11}$	$3.2\ 10^{15}$

Continued on next page

Table 4 – continued from previous page

AS map name	δ	Certificate	Visited 4-tuples	Total number
2008/01/07	5/2	21497, 25533, 31023, 43435	$1.1\ 10^{11}$	$3.1\ 10^{15}$
2008/01/14	5/2	20837, 21497, 25533, 31023	$9.7\ 10^{10}$	$3.1\ 10^{15}$
2008/01/21	5/2	21497, 25533, 31023, 43435	$1.2\ 10^{11}$	$3.2\ 10^{15}$
2008/01/28	5/2	25533, 31023, 43266, 43435	$8.2\ 10^{10}$	$3.1\ 10^{15}$
2008/02/04	5/2	4796, 9326, 23596, 24337	$1.3\ 10^{11}$	$3.1\ 10^{15}$
2008/02/11	5/2	31023, 34265, 43266, 43435	$1.3\ 10^{11}$	$3.2\ 10^{15}$
2008/02/18	5/2	31117, 33851, 42379, 42510	$3.3\ 10^{12}$	$3.3\ 10^{15}$
2008/02/25	5/2	15696, 20545, 42109, 44395	$1.1\ 10^{11}$	$3.3\ 10^{15}$
2008/03/03	5/2	11317, 13856, 23919, 38600	$1.10\ 10^{11}$	$3.3\ 10^{15}$
2008/03/10	5/2	11317, 13856, 23919, 38600	$1.10\ 10^{11}$	$3.4\ 10^{15}$
2008/03/17	5/2	4796, 7588, 9326, 23900	$1.9\ 10^{11}$	$3.4\ 10^{15}$
2008/03/24	5/2	9207, 15577, 29091, 42109	$5.5\ 10^{10}$	$1.3\ 10^{15}$
2008/03/31	5/2	15696, 20545, 42109, 44395	$1.1\ 10^{11}$	$3.3\ 10^{15}$
2008/04/07	5/2	12742, 39863, 43129, 44340	$1.1\ 10^{11}$	$3.3\ 10^{15}$
2008/04/14	2	10044, 24620, 24703, 34620	$4.4\ 10^{13}$	$3.4\ 10^{15}$
2008/04/21	5/2	15577, 17800, 27841, 29599	$1.6\ 10^{11}$	$3.4\ 10^{15}$
2008/04/28	5/2	9326, 23900, 23919, 24514	$2.2\ 10^{11}$	$3.6\ 10^{15}$
2008/05/05	5/2	14085, 29453, 33924, 34771	$2.0\ 10^{11}$	$3.6\ 10^{15}$
2008/05/12	2	12369, 23948, 42109, 43256	$4.4\ 10^{13}$	$3.6\ 10^{15}$
2008/05/19	5/2	7588, 10044, 18362, 38005	$1.3\ 10^{11}$	$3.6\ 10^{15}$
2008/05/26	5/2	12742, 39863, 43129, 44340	$1.0\ 10^{11}$	$3.6\ 10^{15}$
2008/06/02	5/2	15696, 17726, 24475, 24489	$7.6\ 10^{10}$	$6.10\ 10^{14}$
2008/06/09	5/2	12742, 42991, 43129, 44340	$1.2\ 10^{11}$	$3.6\ 10^{15}$
2008/06/16	5/2	9326, 23900, 24487, 38517	$2.0\ 10^{12}$	$3.7\ 10^{15}$
2008/06/23	2	12742, 25747, 42719, 44340	$4.6\ 10^{13}$	$3.6\ 10^{15}$
2008/06/30	2	13548, 34776, 40981, 44340	$4.8\ 10^{13}$	$3.6\ 10^{15}$
2008/07/07	2	4796, 9528, 10044, 25533	$4.5\ 10^{13}$	$3.7\ 10^{15}$
2008/07/14	2	4796, 9528, 10044, 35444	$4.7\ 10^{13}$	$3.7\ 10^{15}$
2008/07/21	5/2	24489, 34776, 35644, 43050	$2.6\ 10^{12}$	$3.9\ 10^{15}$
2008/07/28	5/2	12742, 15458, 39027, 43050	$1.7\ 10^{11}$	$3.7\ 10^{15}$
2008/08/04	2	14085, 16384, 30696, 36269	$1.8\ 10^{10}$	$3.9\ 10^{13}$
2008/08/11	5/2	2614, 20770, 34776, 43930	$1.9\ 10^{11}$	$3.9\ 10^{15}$
2008/08/18	2	20770, 25533, 34639, 34950	$3.6\ 10^{13}$	$2.5\ 10^{15}$
2008/08/25	5/2	12742, 20770, 30790, 34425	$2.0\ 10^{11}$	$3.9\ 10^{15}$
2008/09/01	5/2	12997, 25349, 43030, 43060	$3.2\ 10^{11}$	$4.7\ 10^{15}$
2008/09/10	2	8863, 10044, 12764, 42713	$6.4\ 10^{13}$	$5.2\ 10^{15}$
2008/10/20	5/2	17542, 36970, 36996, 38900	$1.7\ 10^{11}$	$5.4\ 10^{15}$
2008/11/10	5/2	15775, 34881, 41390, 41918	$2.7\ 10^{11}$	$5.5\ 10^{15}$
2008/12/01	5/2	20516, 41540, 42546, 48127	$2.5\ 10^{11}$	$5.6\ 10^{15}$
2008/12/22	2	19956, 22306, 27266, 42136	$7.6\ 10^{13}$	$5.8\ 10^{15}$
2009/01/05	2	8643, 15416, 17641, 29663	$7.5\ 10^{13}$	$5.7\ 10^{15}$
2009/01/10	5/2	18547, 18941, 30952, 47376	$2.5\ 10^{11}$	$5.8\ 10^{15}$
2009/01/22	2	11840, 41641, 42266, 65123	$8.0\ 10^{13}$	$5.8\ 10^{15}$
2009/02/01	2	8643, 10044, 36999, 44036	$7.9\ 10^{13}$	$5.10\ 10^{15}$
2009/02/20	2	15536, 40322, 44036, 45152	$7.8\ 10^{13}$	$6.1\ 10^{15}$
2009/03/11	2	10965, 17542, 28719, 39863	$8.1\ 10^{13}$	$6.2\ 10^{15}$

Continued on next page

Table 4 – continued from previous page

AS map name	δ	Certificate	Visited 4-tuples	Total number
2009/04/29	2	10200, 17739, 43060, 47254	9.1 10^{13}	3.10 10^{15}
2009/05/10	5/2	15440, 15536, 35358, 48053	3.1 10^{11}	6.5 10^{15}
2009/05/20	2	10078, 21274, 34312, 34776	9.6 10^{13}	6.6 10^{15}
2009/06/15	5/2	5530, 24609, 25029, 44784	3.2 10^{11}	6.7 10^{15}
2009/07/01	2	25035, 25503, 43030, 43934	9.2 10^{13}	6.8 10^{15}
2009/07/20	2	17641, 34543, 41281, 47791	9.6 10^{13}	6.8 10^{15}
2009/08/10	5/2	5379, 6906, 12345, 15507	2.9 10^{11}	6.9 10^{15}
2009/08/30	5/2	5379, 6906, 12345, 15507	2.6 10^{11}	6.10 10^{15}
2009/09/20	5/2	3344, 20545, 27933, 42991	3.1 10^{11}	7.1 10^{15}
2009/10/20	2	10097, 11001, 17668, 43030	1.0 10^{14}	7.4 10^{15}
2009/11/20	2	10044, 20803, 30928, 48302	9.8 10^{13}	7.5 10^{15}
2009/12/15	2	4382, 19616, 38140, 45146	1.1 10^{14}	7.8 10^{15}
2010/01/20	2	10200, 18061, 25319, 38140	5.6 10^{13}	8.0 10^{15}
2011/01/16	2	10044, 28750, 43111, 44379	4.4 10^{13}	1.2 10^{16}
2012/06/01	2	132106, 24489, 41068, 58445	1.0 10^{14}	1.9 10^{16}

C Hyperbolicity of some DIMES AS maps

Table 5 lists the hyperbolicity of all DIMES AS maps since 2007 [27], with the corresponding certificates. We have also reported in this table the number of visited 4-tuples for Algorithm 2 with and without good-core decomposition, as well as the total number of 4-tuples in the largest biconnected components of these graphs. We observe that Algorithm 2 out-performs the basic algorithm by a factor of at least 10^4, and that the good-core decomposition provides in average a extra factor 10.

Table 5: Hyperbolicity of DIMES AS maps since 2007.

AS map name	δ	Certificate	Visited 4-tuples		Total number
			With	Without	of 4-tuples
ASEdges1_2007	2	9814, 10063, 26415, 38019	$3.9\ 10^{10}$	$4.2\ 10^{11}$	$8.7\ 10^{14}$
ASEdges2_2007	2	11546, 24398, 38012, 38018	$6.9\ 10^{10}$	$6.5\ 10^{11}$	$6.8\ 10^{14}$
ASEdges3_2007	2	8809, 9791, 17423, 23647	$6.4\ 10^{10}$	$8.5\ 10^{11}$	$1.7\ 10^{15}$
ASEdges4_2007	2	80, 6820, 8701, 9814	$9.4\ 10^{10}$	$1.1\ 10^{12}$	$1.2\ 10^{15}$
ASEdges5_2007	2	2884, 7588, 10755, 40127	$3.3\ 10^{10}$	$4.4\ 10^{11}$	$1.8\ 10^{15}$
ASEdges6_2007	2	6764, 8517, 9812, 15836	$5.6\ 10^{10}$	$7.3\ 10^{11}$	$1.4\ 10^{15}$
ASEdges7_2007	2	10022, 27691, 28871, 32341	$1.0\ 10^{10}$	$2.9\ 10^{10}$	$2.5\ 10^{15}$
ASEdges8_2007	2	8973, 29389, 34870, 35349	$3.1\ 10^{9}$	$4.7\ 10^{10}$	$2.3\ 10^{15}$
ASEdges9_2007	2	841, 9575, 10550, 22684	$1.3\ 10^{10}$	$1.9\ 10^{11}$	$2.3\ 10^{15}$
ASEdges10_2007	2	80, 8511, 34870, 39608	$2.0\ 10^{10}$	$2.5\ 10^{11}$	$1.9\ 10^{15}$
ASEdges11_2007	2	80, 5719, 20661, 35349	$1.4\ 10^{10}$	$1.4\ 10^{11}$	$1.7\ 10^{15}$
ASEdges12_2007	2	13079, 20661, 34205, 34400	$1.1\ 10^{10}$	$1.1\ 10^{11}$	$1.6\ 10^{15}$
ASEdges1_2008	2	7588, 9781, 16905, 37977	$3.6\ 10^{10}$	$3.9\ 10^{11}$	$1.3\ 10^{15}$
ASEdges2_2008	2	18353, 38018, 38319, 41050	$6.4\ 10^{10}$	$4.4\ 10^{11}$	$3.9\ 10^{14}$
ASEdges3_2008	2	9270, 10063, 19008, 38511	$3.6\ 10^{10}$	$4.3\ 10^{11}$	$1.3\ 10^{15}$
ASEdges4_2008	2	7588, 24040, 24287, 42918	$3.3\ 10^{10}$	$4.2\ 10^{11}$	$1.2\ 10^{15}$
ASEdges5_2008	2	8226, 12497, 12987, 37971	$3.1\ 10^{10}$	$3.6\ 10^{11}$	$1.2\ 10^{15}$
ASEdges6_2008	2	28947, 29394, 31094, 34875	$1.6\ 10^{10}$	$1.4\ 10^{11}$	$1.3\ 10^{15}$
ASEdges7_2008	2	5572, 23030, 30214, 33962	$3.7\ 10^{10}$	$9.1\ 10^{10}$	$3.5\ 10^{15}$
ASEdges8_2008	2	5572, 23030, 30214, 33962	$2.5\ 10^{10}$	$1.2\ 10^{12}$	$4.4\ 10^{15}$
ASEdges9_2008	2	5572, 23030, 30214, 33962	$6.0\ 10^{10}$	$2.7\ 10^{11}$	$4.5\ 10^{15}$
ASEdges10_2008	2	2588, 12261, 24042, 24475	$2.0\ 10^{10}$	$8.0\ 10^{10}$	$4.6\ 10^{13}$
ASEdges11_2008	2	3836, 36996, 36982, 38600	$4.7\ 10^{10}$	$4.6\ 10^{11}$	$2.3\ 10^{15}$
ASEdges12_2008	2	2012, 33852, 40981, 42688	$1.2\ 10^{10}$	$1.4\ 10^{11}$	$3.0\ 10^{15}$
ASEdges1_2009	2	10063, 25253, 29673, 30947	$1.6\ 10^{10}$	$1.3\ 10^{11}$	$2.5\ 10^{15}$
ASEdges2_2009	2	4667, 8701, 17752, 38018	$5.0\ 10^{9}$	$4.1\ 10^{10}$	$2.8\ 10^{15}$
ASEdges3_2009	2	41831, 43040, 47360, 47392	$6.1\ 10^{9}$	$7.8\ 10^{10}$	$3.3\ 10^{15}$
ASEdges4_2009	2	11182, 21719, 47392, 47732	$1.4\ 10^{10}$	$1.6\ 10^{11}$	$2.5\ 10^{15}$
ASEdges5_2009	2	34205, 34974, 44674, 48858	$2.8\ 10^{9}$	$2.1\ 10^{10}$	$4.7\ 10^{15}$
ASEdges6_2009	2	12304, 15895, 47326, 48008	$1.4\ 10^{10}$	$5.9\ 10^{10}$	$5.0\ 10^{15}$
ASEdges7_2009	2	2148, 9519, 21418, 47645	$1.6\ 10^{10}$	$2.7\ 10^{10}$	$4.7\ 10^{15}$
ASEdges8_2009	2	4796, 10063, 17585, 35518	$1.6\ 10^{9}$	$1.6\ 10^{10}$	$4.9\ 10^{15}$
ASEdges9_2009	2	8462, 8940, 25432, 48965	$5.4\ 10^{9}$	$1.9\ 10^{11}$	$4.5\ 10^{15}$
ASEdges10_2009	2	4667, 6342, 10131, 17484	$5.5\ 10^{9}$	$5.6\ 10^{10}$	$4.4\ 10^{15}$
ASEdges11_2009	2	13247, 29259, 43910, 47310	$2.3\ 10^{11}$	$5.4\ 10^{10}$	$4.0\ 10^{15}$

Continued on next page

Table 5 – continued from previous page

AS map name	δ	Certificate	Visited		Total
ASEdges12_2009	2	16333, 47360, 48011, 48265	$5.3\ 10^{10}$	$7.2\ 10^{10}$	$3.7\ 10^{15}$
ASEdges1_2010	2	24783, 32277, 34205, 44674	$6.2\ 10^{9}$	$6.5\ 10^{10}$	$4.4\ 10^{15}$
ASEdges2_2010	2	7588, 10004, 17823, 38550	$7.6\ 10^{9}$	$5.1\ 10^{10}$	$4.6\ 10^{15}$
ASEdges3_2010	2	15236, 28509, 42688, 65090	$3.3\ 10^{9}$	$4.5\ 10^{10}$	$4.9\ 10^{15}$
ASEdges4_2010	2	9038, 28666, 29238, 48832	$5.6\ 10^{9}$	$6.0\ 10^{10}$	$4.8\ 10^{15}$
ASEdges5_2010	2	14328, 19915, 22512, 33738	$5.5\ 10^{10}$	$6.2\ 10^{10}$	$5.2\ 10^{15}$
ASEdges6_2010	2	9082, 45012, 45844, 47442	$6.2\ 10^{10}$	$2.7\ 10^{11}$	$4.8\ 10^{15}$
ASEdges7_2010	2	8601, 39644, 48759, 50164	$2.4\ 10^{10}$	$4.6\ 10^{11}$	$4.6\ 10^{15}$
ASEdges8_2010	2	16030, 28951, 29015, 35444	$9.4\ 10^{10}$	$3.9\ 10^{11}$	$5.7\ 10^{15}$
ASEdges9_2010	2	8395, 18239, 23850, 42136	$1.5\ 10^{10}$	$1.1\ 10^{11}$	$5.2\ 10^{15}$
ASEdges10_2010	2	10150, 18252, 23940, 38195	$2.7\ 10^{10}$	$8.9\ 10^{10}$	$5.7\ 10^{15}$
ASEdges11_2010	2	10282, 16292, 196840, 39425	$5.6\ 10^{9}$	$5.2\ 10^{10}$	$5.3\ 10^{15}$
ASEdges12_2010	2	8557, 18352, 38437, 55454	$2.5\ 10^{10}$	$7.6\ 10^{11}$	$5.2\ 10^{15}$
ASEdges1_2011	2	2503, 18088, 36790, 38043	$9.2\ 10^{10}$	$7.9\ 10^{11}$	$5.9\ 10^{15}$
ASEdges2_2011	2	12576, 29032, 29039, 36892	$3.8\ 10^{10}$	$1.7\ 10^{11}$	$6.8\ 10^{15}$
ASEdges3_2011	2	15633, 16030, 25247, 31483	$9.0\ 10^{10}$	$1.5\ 10^{11}$	$4.9\ 10^{15}$
ASEdges4_2011	2	8193, 34557, 41904, 43246	$1.6\ 10^{9}$	$2.3\ 10^{10}$	$5.0\ 10^{15}$
ASEdges5_2011	2	10207, 11879, 262815, 28174	$1.4\ 10^{10}$	$6.2\ 10^{10}$	$4.8\ 10^{15}$
ASEdges6_2011	2	841, 6853, 8338, 25103	$4.0\ 10^{10}$	$6.3\ 10^{11}$	$4.3\ 10^{15}$
ASEdges7_2011	2	2874, 29498, 44192, 45289	$1.1\ 10^{11}$	$8.9\ 10^{11}$	$4.7\ 10^{15}$
ASEdges8_2011	2	23817, 25669, 33516, 40335	$3.9\ 10^{10}$	$1.0\ 10^{11}$	$4.1\ 10^{15}$
ASEdges9_2011	2	32017, 37187, 37245, 49283	$6.8\ 10^{9}$	$8.7\ 10^{10}$	$4.0\ 10^{15}$
ASEdges10_2011	2	12705, 29304, 34875, 53603	$1.4\ 10^{10}$	$2.0\ 10^{11}$	$3.6\ 10^{15}$
ASEdges1_2012	2	10110, 18021, 35794, 38305	$2.6\ 10^{10}$	$2.5\ 10^{11}$	$3.1\ 10^{15}$
ASEdges2_2012	2	24490, 28249, 45344, 45410	$2.7\ 10^{10}$	$3.2\ 10^{11}$	$3.7\ 10^{15}$
ASEdges3_2012	2	7491, 41579, 47654, 56384	$3.3\ 10^{10}$	$4.4\ 10^{11}$	$3.7\ 10^{15}$
ASEdges4_2012	2	24337, 38296, 38738, 47652	$4.6\ 10^{10}$	$5.6\ 10^{11}$	$3.4\ 10^{15}$

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399